Networked Multi-robot Collaboration in Cooperative–Competitive Scenarios Under Communication Interference

Yaowen Zhang[1], Dianxi Shi[1,2(✉)], Yunlong Wu[1,2(✉)], Yongjun Zhang[3], Liujing Wang[2], and Fujiang She[1]

[1] Artificial Intelligence Research Center (AIRC), National Innovation Institute of Defense Technology (NIIDT), Beijing 100071, China
{dxshi,ylwu1988}@nudt.edu.cn
[2] Tianjin Artificial Intelligence Innovation Center (TAIIC), Tianjin 300457, China
[3] National Innovation Institute of Defense Technology (NIIDT), Beijing 100071, China

Abstract. In this paper, we consider a scenario where a team of predator robots collaboratively survey an area for preventing the invasion from opponent robots. In this scenario, the predator robots can share the sensing information of the prey robots through wireless communication. In order to constrain the surveillance performance of the predator robots, besides the prey robots, some interfering robots are added to break the communication connectivity between the predator robots. This is a typical "cooperative–competitive" decision problem involving multiple optimization variables from electromagnetic and geographic domains, which makes it challenging to solve. For this problem, we first propose the perception and communication models of the robots. Then, with these models, we formulate the problem and adopt multi-agent reinforcement learning (MARL) to solve it. Furthermore, considering the long training-time cost of traditional MARL, we propose a scenario curriculum learning (SCL) training strategy, which can reduce the computation time and improve the performance by evolving the scenarios from simplicity to complexity. The effectiveness of the proposed method is verified by the analysis and simulation results. The results show that the SCL strategy can reduce the training time by 13%.

Keywords: Electromagnetic and geographic domains · Multi-agent reinforcement learning · Scenario curriculum learning

1 Introduction

With the rapid development of artificial intelligence and automation technology, robots are widely used in various fields such as industries, safety, military, and

Supported by the National Key Research and Development Program of China under Grand No. 2017YFB1001901, the Key Program of Tianjin Science and Technology Development Plan under Grant No. 18ZXZNGX00120 and the National Natural Science Foundation of China under Grant No. 61906212.

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 349, pp. 601–619, 2021.
https://doi.org/10.1007/978-3-030-67537-0_36

scientific research [1]. Compared with single-robot systems, multi-robot systems (MRS) can effectively improve the execution efficiency of tasks through collaboration and have greatly enhanced survivability and adaptability in complex environments [2]. Multi-robot collaboration often relies on information sharing and interaction between individuals through networking [3]. However, in a complex environment, more diversity and confrontation exist. In particular, in the case of active signal jamming, communication connectivity will face challenges. Therefore, how to ensure multi-robot coordination in complex environments is a challenge in the robotics field.

An intuitive way to model the behavior of an MRS is to predefine the action rules for each robot [4]. However, enumeration of the entire situations is difficult. Moreover, the action of each robot further results in a continuously changing environment [5]. Therefore, it's challenging to solve the decision problem of multi-robot for "cooperative–competitive" environment in "electromagnetic–geography" multi-domain. Because it need to be involved multiple optimization variables from electromagnetic and geographic domains from practicality.

In this paper, We first model the perception and communication of robots, then describe the problem with constraints as Markov decision process (MDP), and use Multi-Agent Reinforcement Learning (MARL) to solve it. The current MARL solution concept is based on centralized training and distributed execution mode. This mode can effectively improve the adaptability and convergence of the algorithm to complex environments by considering the action policies of all agents [6]. And it's widely used to solve decision problems in multi-agent systems. Some typical algorithms include DDPG [7], MADDPG [8], and COMA [9]. The advantage of the distributed learning mode for multi-agent system is that each agent in the system, having its own independent action policy, can optimize its own behavior by interacting with other agents. We analyze and model this multi-domain problem and aim to solve it using the MADDPG algorithm, because this algorithm has the advantages of distributed training and good expansibility.

In addition, each agent in MARL regards other agents as part of the environment and it observes and interacts with the environment to obtain reward, thereby the strategy of the multi-agent system could be updated [8,9]. With the increasing number of agents, the computational complexity and run-time of MARL increases exponentially [6]. In a multi-agent system, every agent aims to learn the best response to the behavior of others. If the other agents also adapt their strategy, the learning target moves, that is, the trained system strategy would be non-stationary [10], and efficiency will also be far away from expectations. In order to improve the stationary and efficiency of the algorithm, we propose a training method based on complex scenario curriculum learning (SCL).

However, current researches on MARL focus on relatively simple environments [4]. Many standardized environments such as the Arcade Learning Environment [11] are simple plane environment for the RL algorithm benchmark. Other platforms such as StarCraft Multiagent (a real-time strategy game) [12] and Hanabi (a multiplayer card game) [13] also involve single domains. To the

best of our knowledge, this is the first time reinforcement learning is used to solve the problem of a MRS under "cooperative–competitive" in an "electromagnetic–geography" multi-domain environment. The following are the main contributions of this paper:

(1) We build the communication and perception models of the robots involving in the scenario. In the models, we further consider the communication interference effects from the interferencing robot.
(2) We formulate the "cooperative-competitive" problem which involves the cooperation between homogeneous robots and competition between heterogeneous robots.
(3) We propose a training method based on transfer learning for improving training efficiency and effectiveness. We adopt different scenarios to train the policies of the robots and can obtain a better cooperation strategy than the directly training.
(4) Numerical results demonstrate that the optimized collaboration model could be trained by DRL and we analyze the difference and advantage between two algorithms. We further prove the improvement of SCL in complex scenarios.

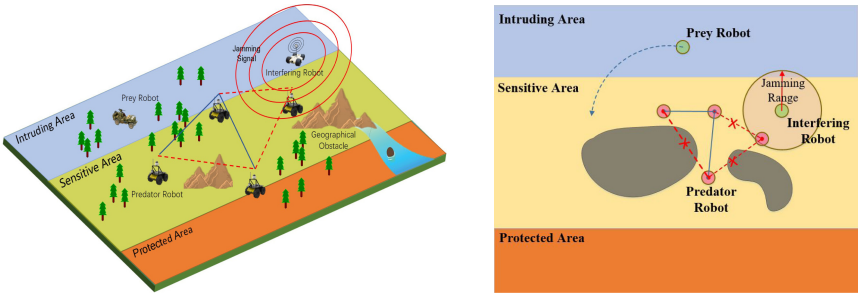
The rest of the paper is organized as follows. Section 2 presents the scenario modeling, and Sect. 3 presents the solution by implement MADDPG with depth-first search algorithm (DFS). In Sects. 4 and 5, we introduce SCL and describe the transfer process step by step. Section 6 describes the simulation experiments conducted to solve and verify method optimization. Finally, we conclude the study in Sect. 7.

2 Modeling

2.1 Problem Description

The scenario is shown in Fig. 1(a). A team of homogeneous predator robots are responsible for patrolling the sensitive area and maintaining communication connectivity among robots. The task of prey robots, acting as an opponent agent, is to break through the sensitive area from the intruding area to invade into the protected area. And heterogeneous robots, such as the predator robot and prey robot in the geospace, as well as the predator and interfering robot in the electromagnetic field, competitive to achieve own goals. The round ends when the prey robots are captured by the predator robots while the prey robots are in the protected area. Interfering robots (also opponent agents) disturb the communication links between predator robots.

Based on the above description, the problem can be modeled as an MDP [14] and defined as a tuple $\{S, A, R, P, \gamma\}$, where S be the state set for the MRS; A_i and O_i be the action and observation space of agent i. If an agent has limited perception ability and the environment state changes dynamically, the environment is partially observable for the agent. Therefore, we could define transition probabilities between states $P : S \times A_1 \times ... \times A_N \mapsto S$. The reward



(a) Multi-agent in multi-domain scenario diagram.

(b) The scenario modeling diagram.

Fig. 1. There are 4 homogeneous robots patrol the sensitive area collaboratively, and heterogeneous robots confrantation described in Fig. 1(a). We can model this scenario simply as Fig. 1(b).

function for an agent is $r_i : S \times A_i \mapsto \mathbb{R}$ and the observation is $\mathbf{o_i} : S \mapsto O_i$. The goal of agent i is to learn the optimal policy π_i^* that maximizes the expected return with a discount factor γ within a time range T:$R_i = \mathbb{E}_{s_i,a_i}\left[\sum_{t=0}^{T} \gamma^t r_i^t\right]$.

2.2 Robot Model

In this section, we first introduce the notation (Table 1) involved in our model, and then describe three types of robots in our problem: predator, prey, and interfering robots. The relationship between them is as follows: cooperative relationship between predator robots, meanwhile the prey and interfering robots also cooperate to confront predator robots. Competitive relationship between predator robots and opponent (prey and interference) robots. We consider the scenario is a two-dimensional planar environment. The position of robot i can be expressed as $p_i = [x_i, y_i], x_i, y_i \in \mathbb{R}$, and robot actions can be expressed as $a_i = [v_i, \theta]$ $(a_i \in A_i, v_i \in [V_{\max}, V_{\min}], \theta \in [0, 2\pi))$, where $V_{\max}$ and $V_{\min}$ are maximum and minimum speeds and θ is the orientation angle.

Table 1. Notation table.

Symbol	Explanation
$\mathbf{o}$	Observation vector
$\mathbf{u}$	Perception vector
$\mathbf{c}$	Communication vector
$\mathbf{r}_{i,c}$	Communication radius of predator robot i
U_{c}^i	Joint-communication scope of predator robot i
U_{s}^i	Uni-sensor scope of predator robot i
$r_i^{(k)}$	$k-$th reward of robot i

Predator Robots. The task of a predator robot is to capture preys within its own observable scope, and the perception vision can be shared between connective predators via communication. Therefore, we define their behavior to comprise collaboration, patrol, and active connection. Observation vector $\mathbf{o}_{i,t} \in O_{i,t}$ of predator i at time t consists of two parts: perception vector $\mathbf{u}_{i,t}$ and communication vector $\mathbf{c}_{i,t}$. $\mathbf{u}_{i,t}$ is a list of detected opponent robots, and $\mathbf{u}_{i,t} = \left[u_{i,t}^1, ..., u_{i,t}^j, ..., u_{i,t}^P\right]$, where P is the number of opponents. $\mathbf{c}_{i,t}$ is a list of connective teammates (predators), and $\mathbf{c}_{i,t} = \left[c_{i,t}^{P+1}, ..., c_{i,t}^{P+k}, ..., c_{i,t}^{P+I}\right]$, where I is the number of interconnected predators. $I + P \leq N - 1$ holds, where N is the total number of robots.

We assume the perception scope as a circle with radius r_s for $\mathbf{u}_{i,t}$, and distance between predator i and opponent j as the Euclidean distance $d_{i,j} = ||p_i - p_j||$. The element u_i^j in vector $\mathbf{u}_i$ could be defined as

$$u_i^j = d_{i,j}, d_{i,j} \leq r_\mathrm{s} \tag{1}$$

In addition, the communication scope is modeled as a circle with radius r_c for $\mathbf{c}_{i,t}$. In practice, communication scope is larger than that of perception, so we have $r_\mathrm{s} < r_\mathrm{c}$. The element c_i^{P+k} $(k \in [1, I])$ in vector $\mathbf{c}_i$ could be defined as

$$c_i^{P+k} = d_{i,k}, d_{i,k} \leq r_\mathrm{c} \tag{2}$$

In order to maintain connectivity under EMI signal, predators adopt two modes: conventional communication (lower power for energy saving) and strong communication (high-power and strong directivity, consumes more energy, and resists interference to a certain extent) [15]. The two modes can be modeled as circles with different radii. Interfering robot l has a circular interference area with a radius of r_o. For predator robots i and k, the communication radius of robot i is

$$r_{i,\mathrm{c}} = \begin{cases} r_\mathrm{h}, d_{i,k} \leq d_{i,l} \text{ or } d_{i,l} \geq r_\mathrm{o} \\ r_\mathrm{l}, d_{i,k} > d_{i,l} \text{ and } d_{i,l} < r_\mathrm{o} \end{cases} \tag{3}$$

Mode selection of the current predator robot according to the relationship of $d_{i,l}$ and $d_{i,k}$ (robot k is the nearest collaborative predator) [16]. When robot i is in the signal jamming area and $d_{i,k} \leq d_{i,l}$, the communication will be jammed due to EMI suppression by the interfering robot l. Then, the strong communication mode is on and the radius $r_{i,\mathrm{c}}$ is r_h. In other conditions, the conventional communication mode is maintained and radius $r_{i,\mathrm{c}}$ is r_l. For ensuring a certain energy consumption, the strong communication mode with high power consumption has smaller radius than the conventional mode, i.e., $r_\mathrm{h} < r_\mathrm{l}$.

The communication scope of robot i can be expressed as $\sigma_i = \{p_x | \forall d_{i,x} \leq r_{i,\mathrm{c}}\}$, where x is the index of discrete geospatial. An arbitrary predator k satisfying $p_k \in \sigma_i$ can be used as the mobile relay of robot i, and joint-communication scope of robot i can be expanded to

$$U_\mathrm{c}^i = \{\bigcup_j \sigma_j | \forall d_{i,j} \leq r_{i,\mathrm{c}}, p_j \in \sigma_i, \forall j \in [1, N]\} \tag{4}$$

By applying DFS [17], we can obtain the interconnected predator set G^i about predator i recursively. Based on networking group G^i, the uni-sensor scope can be expanded to

$$U_{\mathrm{s}}^i = \left\{ p_x | \forall d_{j,x} \leq r_{i,\mathrm{s}}, j \in G^i \right\} \tag{5}$$

The predator accumulation reward $R_{i,t}$ is sum by four rewards of obstacle avoidance $R_i^{(0)}$, communication maintenance $R_i^{(1)}$, task $R_i^{(2)}$ and patrol $R_i^{(3)}$. We define obstacles as the set of location points occupied by an area on a two-dimensional plane $C_{\mathrm{obs}} \subset \mathbb{R}^2$. $R_{\mathrm{u}}^{(0)}$ is the obstacle detection reward and we have $R_i^{(0)} = R_{i,\mathrm{u}}^{(0)} + R_{i,\mathrm{w}}^{(0)}$; $R_{i,\mathrm{u}}^{(0)} = \sum_{t=0}^{T} \gamma^t r_{i,\mathrm{u}}^{(0)}$ is given by

$$r_{i,\mathrm{u}}^{(0)} = \begin{cases} -a, p_i \in C_{\mathrm{obs}} \\ 0, p_i \notin C_{\mathrm{obs}} \end{cases} \tag{6}$$

$R_{i,\mathrm{w}}^{(0)}$ is the reward for the joint communication group of predators repelling obstacles and $R_{i,\mathrm{w}}^{(0)} = \sum_{t=0}^{T} \gamma^t r_{i,\mathrm{w}}^{(0)}$ is given by

$$r_{i,\mathrm{w}}^{(0)} = \begin{cases} -b, C_{\mathrm{obs}} \cap U_{\mathrm{c}}^i \neq \varnothing \\ 0, C_{\mathrm{obs}} \cap U_{\mathrm{c}}^i = \varnothing \end{cases} \tag{7}$$

Where $a, b > 0$, p_i is position of robot i, and In the 2-D plane, the joint communication group of interconnected predators aims to avoid obstacles to maintain a good connectivity.

We define $R_i^{(1)} = \sum_{t=0}^{T} \gamma^t r_i^{(1)}$ in terms of the communication vector $\mathbf{c}_i$ as

$$r_i^{(1)} = \exp\left(k^{(1)} \max \mathbf{c}_i\right) - 1 \tag{8}$$

Where $k^{(1)} < 0$ is the scale factor. For disconnected predators, the current robot can actively attempt to reconnect and form a group again.

When a prey robot j's position satisfies $p_j \in \bigcup_{i=1}^{N} U_{\mathrm{s}}^i$, accumulation task reward $R_i^{(2)} = \sum_{t=0}^{T} \gamma^t r_i^{(2)}$ formulate in terms of the perception vector $\mathbf{u}_i$ as

$$r_i^{(2)} = k^{(2)} \min \mathbf{u}_i \tag{9}$$

Where $k^{(2)} < 0$ is the scale factor. The predators chase the prey that is within the perception scope.

For limited observable scope range, predators need patrol in the prescribed area if there has no prey robot appeared in the uni-sensor scope. We have $R_i^{(3)} = \sum_{t=0}^{T} \gamma^t r_i^{(3)}$ as

$$r_i^{(3)} = 1, p_i \in X_i \tag{10}$$

Prey Robot. We assumed that prey robots have global observation, so they could obtain every robot position in the environment. The observation vector of prey robot j is $\mathbf{o}_j = [d_{j,1}, ..., d_{j,k}, ..., d_{j,N}]$. Accumulation reward $R_{j,t}$ for prey

robot j sum by three rewards of obstacle avoidance $R_j^{(0)}$, confrontation $R_j^{(1)}$ and task $R_j^{(2)}$. Here, obstacle avoidance reward $R_j^{(0)}$ is similar to that in the predator robot model.

Confrontation reward $R_j^{(1)} = \sum_{t=0}^{T} \gamma^t r_j^{(1)}$ is defined for keeping away from predators during intrusion. Relationship with $\mathbf{o}_i$ could be expressed through scale factor $\rho^{(1)} > 0$ as

$$r_j^{(1)} = \rho^{(1)} \min \mathbf{o}_i \tag{11}$$

We use function $g(x)$ represents the relationship between robot j position and task reward $R_j^{(2)} = \sum_{t=0}^{T} \gamma^t r_j^{(2)}$. The depth of prey j position is linearly increasing with the reward.

$$r_i^{(2)} = g^{(2)}(p_i) \tag{12}$$

Interfering Robot. The main task of interfering robots l is to interfere with as many predator robots as possible while they avoid obstacles (the reward for avoiding obstacles, $R_l^{(0)}$, is same as that of predators). Therefore, we have

$$R_l = R_l^{(0)} + \sum_{t=0}^{T} \gamma^t ||X^{(i)}||, X^{(i)} = \{i | \forall d_{l,i} \leq r_o, i \in [1, N]\} \tag{13}$$

2.3 Problem Model

Based on the above constraints, we can model this problem for arbitrary robot i,

$$\begin{aligned} \max_{\pi_i} \ & J_i = \mathbb{E}_{s_i, a_i} \left[R_{i,t} \right] \\ \text{s. t. } & o_{i,t} \in O_{i,t}, \\ & a_{i,t} \in A_{i,t}, \\ & t \in [1, T]. \end{aligned} \tag{14}$$

The optimization goal J_i is the sum of cumulative rewards of the infinite horizon and R_i obtained by the interaction of agent with the environment. The optimization variable for this problem is each agent's action policy π_i. The action vector $a_{i,t}$ of agent i at time step t should be satisfied with the behavior policy according to the environment observation $o_{i,t}$ under the constraint of its action space A_i. Our objective is to determine an optimal behavior policy π_i that satisfies the constraints of multi-agent scenarios, so that the objective function can reach the expected maximum in an infinite time range.

3 Solution

3.1 Scenario Hypothesis

For the simulation of our model operation, the following constraints are considered: (1) delay or bandwidth in communication and the error of physical coordinates caused by sensors are ignored. Thus, once an agent is detected, its coordinates can be accessed immediately; (2) unlike [18], we do not use any differential dynamic model for agents; (3) any prey agent in the attack scope of predator agents is considered to be destroyed immediately.

3.2 System Model

We solve various scenarios with a specific actor–critic DRL algorithm framework. The entire system is shown in Fig. 2. The actor network is used to calculate the actions of the current agent according to the states observed from the environment. A critic network is used to evaluate the computation results of the actor network. The critic network has observation information of all agents, thus improving the performance of the mixed cooperative–competitive behavior. A replay buffer pool $\mathcal{D}$ is minibatch for collecting experience of tuples (S, A, R, S') from the environment. In the training process, the input of the actor network is its own observation value about the environment state, while the input of the critic network is not only its own observation value but also other agents' observation values. The critic network calculates the Q value of the state–action pair of the actor network, which is used to update the parameters of actor network.

In this way, using an actor–critic framework, each agent can receive information from other agents for training (i.e., centralized training) and perform actions through its own observation (i.e., decentralized execution). Therefore, each agent can optimize its own behavior policy through the information of other agents. For the trained model, each agent can use the action calculated by the actor network to interact with the environment. Even if an agent has only partial state information, it can still make appropriate decisions to perform actions.

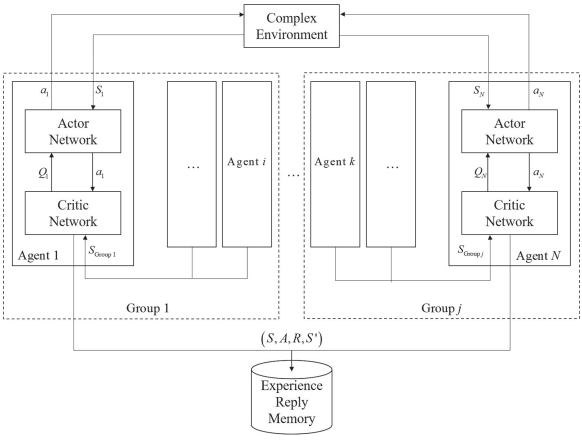


Fig. 2. Actor–critic framework of collaborative agent in networking group.

We assume a set of continuity strategies for all agents $\mu = \{\mu_1, \mu_2, \ldots, \mu_N\}$. If each μ_i corresponds to a parameter vector θ_i in network, then the parameterization strategy mapped to the network can be expressed as $\theta = \{\theta_1, \theta_2, \ldots, \theta_N\}$. Then, we obtain the gradient of the expected revenue of agent i as

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{S,a\tilde{}\mathcal{D}}[\nabla_{\theta_i} \mu_i(a_i|o_i) \nabla_{a_i} Q_i^{\mu}(S^{(k)}, a^{(k)})|_{a_i=\mu_i(o_i)}]. \tag{15}$$

Where Q_i^{μ} is the action value function of an agent. Under the condition of limited observation, agent i accesses the actions $a^{(k)} = \{a_i, ..., a_k\}$ of other agents (i.e., the agents form a group). At the same time, agent i obtains the status information $S^{(k)} = \{a_i, ..., a_k\}(k \leq N)$ of other agents in one group. Finally, the Q_i^{μ} value of agent i is output. The replay buffer pool $\mathcal{D}$ contains a tuple $(S, S', a_1, ..., a_N, r_1, ...r_N)$, which records the experience of all agents. Therefore, the value function Q_i^{μ} is updated as

$$\begin{aligned} \mathcal{L}(\theta_i) &= \mathbb{E}_{S,a,R,S'}\left[\left(Q_i^{\mu}(S, a_1, ..., a_N) - y\right)^2\right], \\ y &= r_i + \gamma Q_i^{\mu'}(S', a'_1, ..., a'_N)|_{a'=\mu'(o_j)} \end{aligned} \tag{16}$$

4 Scenario Curriculum Learning

In MADDPG, a complex environment not only leads to a huge amount of computation, but multi-agent also bring stability issues [10]. Curriculum learning (CL) is defined as a machine learning concept and is designed to improve the performance for transfer learning. In [19], CL was first combined with RL. One major direct application of CL in RL is to deal with complex tasks [20–22]. In CL, the goal is to improve the final asymptotic performance or decrease the computation time by generating a series of tasks. Tasks can be trained individually before progressing to learning on the final task [23].

However, most existing studies (such as those mentioned above) focus on single agents on CL. Although some existing approaches consider CL in a multi-agent system, they utilize CL in an extremely simple manner. Moreover, a single environment is a contrast to the environment considered in our article in that the number of agents and sparse rewards are constant. We propose a multi-agent CL named complex scenario curriculum learning (SCL) as shown in Fig. 3.

SCL solves the non-stationary and multi-agent training effect by starting from learning a simple multi-agent scenario and gradually increasing the number of agents and complication to finally learn the target task. Two kinds of transfer method are proposed across different order and training parameters, which can boost the performance of training on the win rate.

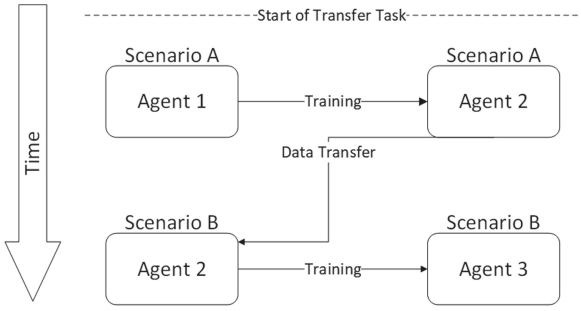


Fig. 3. Agent training in different scenarios.

Figure 3 shows the agent transfer process in different scenarios. First, the order of multi-source task transfer training is determined according to the scene complexity function ϕ . In the initial scene A, N agents are trained for t_1 iterations until approximate convergence, and then the strategy of agents is expressed as $\pi^1_{t_1} = \{\pi^1_1, \pi^1_2, \ldots, \pi^1_N\}$. Then, the above strategy is taken into scene B as the initial condition. Afterwards, agents are trained for t_2 iterations until approximate convergence. Likewise, the model finally converges well in the final scenario. Its algorithm described as Algorithm 1.

Algorithm 1: Scenario Curriculum Learning

Input: Objective task M_t, scenario complexity m, originating task set $\mathcal{M}_s = \{M_0, M_1, ...\}$

Output: Solution of objective task M_t, $\mathcal{H}$

Generate curriculum tasks sequence $\mathbb{O} \leftarrow orderTasks(\mathcal{M}_s, m)$;

while $M_i \subseteq \mathbb{O}$ **do**

  $\mathcal{K}_i \rightarrow \mathcal{H}_i$;

  $\mathcal{H}_i \times \mathcal{K}_{i+1} \rightarrow \mathcal{K}_{transfer}$;

  $\mathcal{K}_{transfer} \times \mathcal{K}_{i+1} \rightarrow \mathcal{H}_{i+1}$;

return $\mathcal{H}_t$;

For single-source task transfer, namely, only a given task source, the agent can extend the prior knowledge learned from the source to the target task [24]. The process of RL modeling of each task is equivalent to an MDP process, so the task space can be represented by a set $\mathcal{M} = \{M_0, M_1, ...\}$.

The process of transfer can be expressed as: input the knowledge of the target task, and output the new solution $\mathcal{H}$ through training in the new scene.

$$\mathbb{A}_{learn} : \mathcal{K} \rightarrow \mathcal{H} \tag{17}$$

$\mathcal{K}$ represents the knowledge space of the source task as prior knowledge and $\mathcal{H}$ represents the solution space of the source task. In the knowledge transfer stage, according to the correlation between the source and the target task, appropriate knowledge is generated. This process can be expressed as follows:

$$\mathbb{A}_{transfer} : \mathcal{K}^n_s \times \mathcal{K}_t \rightarrow \mathcal{K}_{transfer} \tag{18}$$

Where $\mathcal{K}^n_s$ is the knowledge obtained from N source tasks, $\mathcal{K}_t$ is the knowledge of target tasks, and $\mathcal{K}_{transfer}$ is the final target.

In the learning phase, the transferred knowledge and the current task knowledge are used to learn the final solution:

$$\mathbb{A}_{t-learn} : \mathcal{K}_{transfer} \times \mathcal{K}_t \rightarrow \mathcal{H} \tag{19}$$

From Eq. (4.3), we can see that the current task uses $\mathcal{K}_{transfer}$ as additional knowledge when learning; therefore, the transfer algorithm and solution space

dimension should be consistent with those of the target task. Thus, we use the policy π_t^0=$\{\pi_1^0, \pi_2^0, \ldots, \pi_N^0\}$.

which has been trained in source task M_0 after t iterations as the next task's initial solution. It is expressed as

$$\pi_s(p(S_t)) = \sigma[\pi_t^0(S_t)] \tag{20}$$

Where π_t^0 is the initial policy of the target task and π_s is the policy of the source task. During scenario transfer, the correspondence between the state space and action space of source task M_t^0 and target task M_s is $p : S_t \rightarrow S_s$ and $\sigma : A_t \rightarrow A_s$, respectively.

5 Scenarios

5.1 Scenario 1: Global Observation

In this section, we first consider a basic scenario involving two types of roles: the predator robot and the prey robot. Every robot has a global perception and communication scope. The confrontation scene is described in Fig. 4.

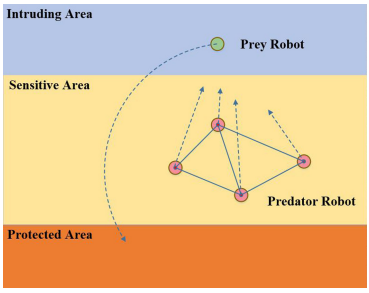
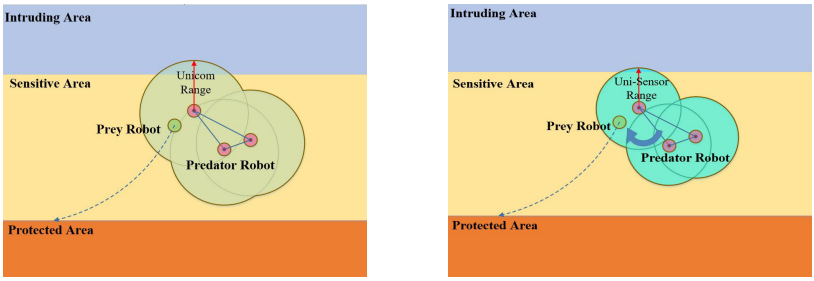


Fig. 4. Predators chase a prey robot synergistically.

In this part, we expect the predator robots to learn the action policy of collaborative encirclement and prey robots to learn the invitation policy by maximizing sum J in finite-horizon t_1 cumulative rewards as Eq. 2.14, and then reach Nash equilibrium in the competition between the two sides.

5.2 Scenario 2: Partial Observation

Based on Subsect. 5.1, we consider that a predator robot has the ability to have a local communication in Fig. 5(a) and partial perception in Fig. 5(b). Without considering communication interference, the constraints of sensing and connecting are, respectively, the same as Eq. (2.1) and Eq. (2.2) in Subsect. 2.2. And the extended areas in joint communication and the uni-sensor are expressed as Eq. (2.4) and Eq. (2.5), respectively. The reward functions of both sides are the same as Eq. (2.8)–Eq. (2.12).



(a) Establish connection. (b) Extend sensor scope.

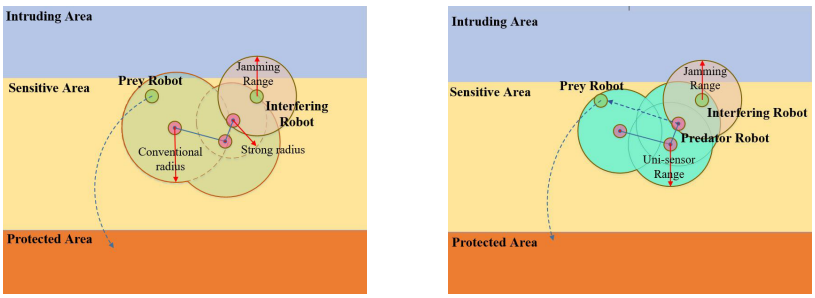
Fig. 5. Interconnective robots form a network group as Fig. 5(a) and extend uni-sensor scope to maximize perception range as Fig. 5(b).

The trained model described in Subsect. 5.1 is taken as the initial input for training the ability of maintaining the communication group based on the hunting strategy of predator robots. The predators in common group can chase the perceived prey robot collaboratively.

5.3 Scenario 3: Communication Under the Signal Jamming

In this part, we introduce the interfering robot and the suppression of the EMI signal transmitted by the interfering agent. Therefore, the radius of the conventional or strong communication mode is modeled as Eq. (2.3) in Subsect. 2.2. The process of this scenario is described in Fig. 6. We deem the scenario 2 trained model in t_2 iteration as the initial condition in scenario 3, and run at t_3 horizon.

In this scenario, we consider competition in electromagnetic domain. Predator which in the EMI area could patrol with conventional mode without joint-perception or switch strong mode to maintain interconnection.



(a) Convert communication mode under signal interference. (b) Joint perception area by mobile relay robot.

Fig. 6. Strong communication mode helps maintain connection when predator robot position is in EMI signal area. Another predator can move closer to satisfied communication condition for maximum joint-perception scope.

5.4 Scenario 4: Collaboration Under the Interference

On the basis of the model trained in Subsect. 5.3 after iteration t_4, we continue to train the behavior of robots in the scene. The modeling detail is consistent with that in Sect. 2 as shown in Fig. 1(b).

The obstacles not only could hinder robots movement but also interrupt communication link. In the geographic field, both robots take action in confrontation on the premise of avoiding obstacles. In the electromagnetic field, competition happens between the two sides with "interference–counteract interference".

6 Simulation

We performed simulations to verify the effectiveness of the models proposed in Sect. 3 and optimization in Sect. 4. The workspace is a 200 m × 200 m square region, and the effective destruction radius of the predator robot is 10 m. The perceptive radius is $r_s = 40$ m, the conventional communication radius is $r_c = r_l = 80$ m, the strong communication radius is $r_c = r_h = 60$ m, and the EMI radius of the interfering robot is $r_o = 40$ m. Moreover, we set the predator robot speed to 10 m/s, the prey robot speed as 12 m/s, and the interfering robot speed as 8 m/s. The simulation platform is a desktop computer equipped with a i7 CPU and a NVIDIA Geforce RTX 2080Ti GPU. We adopt Tensorflow 1.18, Open Gym 0.15 and Python 3.5 for experimental simulation verification.

6.1 Multi-robot in Multi-domain Scenario

In this subsection, we describe the simulation of the scenario described in Sect. 2, and determine the optimal collaboration strategy model between predator robots under confrontation with opponents. For validating our conclusion, we compared the win rates (Fig. 7) and reward results (Fig. 8) obtained by various DRL algorithms applied for this scenario.

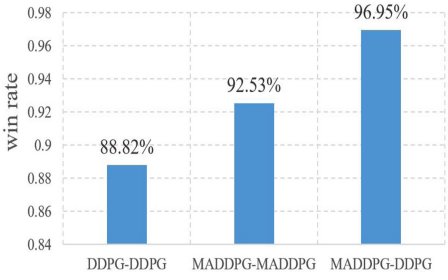


Fig. 7. Win rates for robots trained using various algorithms.

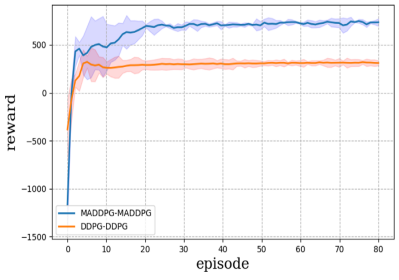


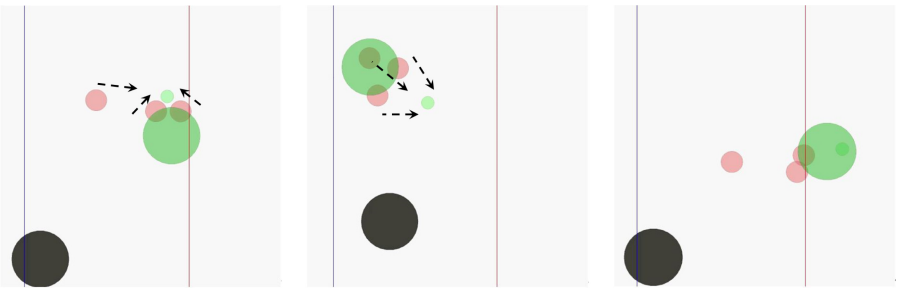
Fig. 8. Reward scores obtained through various DRL algorithms in scenario 4.

We define the win rate as that in 30,000 rounds of confrontation scenarios in Gym simulation environment. For the predator robots winning, they could capture the prey robot which has not reached the protected area.

After 80,000 iterations of training, the left and middle results shown in Fig. 7 indicate that both DDPG and MADDPG algorithms could train the predators to obtain an effective collaborative strategy. The MADDPG model provided a higher win rate than DDPG. Moreover, as shown in Fig. 8, MADDPG trained predator robot acquires more reward in a multi-domain environment.

In Fig. 7, the rightmost results are obtained by a cross-comparison experiment. Here, we trained the network model by both DDPG and MADDPG algorithms: the predator robot imported the model trained by MADDPG, and the opponent robot imported the model trained by DDPG. Compare with the middle result in Fig. 7, for the same DDPG algorithm trains the prey robot, the predator robot trained by MADDPG showed better encircling policy for the intrusion policy of the prey trained by DDPG.

In the simulation results presented in Fig. 9, the red balls represent predator robots, and their size indicates the effective attack range. The green balls represent the opponent robot. The smaller one is prey robot, whereas the larger one is interfering robot. Interfering robot size represents the signal jamming scope. The predator robot can cooperate in the effective range of communication. At the same time, the prey robots wait for the opportunity to assault the protected area under the cover of the interfering robots. All robots can identify and avoid obstacles well. Therefore, it can be proved that MADDPG can solve the networked multi-robot in competitive scenario under interference.



(a) Three predator robots surrounding prey robot.

(b) Pursuit as a formation under EMI signal.

(c) Prey assaults with interferencing robot covering.

Fig. 9. We simulated scaled scenario of Sect. 2. Protected area, sensitive area and intruding area are divide by lines from left to right. Obstacles appearing randomly are introduced, and all robots movement with avoidance behavior.

6.2 SCL Optimization

The goal of SCL is to improve the learning performance of an agent in the target task. The evaluation indicators of the learning performance can be measured

with three aspects [25] that are learning speed improvement, improvement of jumpstart and asymptotic performance. Since the simulation involves the confrontation scenario between the two sides, we should not only compare the above three indicators but also compare the win rate under different scenarios, so as to prove the stability of scenario transfer. In addition, we prove the effectiveness of the proposed method in terms of the computation time.

Based on the above 5 indicators, we design two different scenario transfer cases to compare the impact of different scenario transfer methods on the transfer results. According to the sequence of substep training and the size of replay buffer, symmetric SCL (S-SCL) and asymmetric SCL (A-SCL) are proposed to verify the impact of those indicators on the transfer effect. Meanwhile, from the simulations on the scenarios described in Subsect. 5.4, we can observe the effect of different agent numbers on the transfer effect.

Symmetric SCL. In this subsection, the curriculum is designed as scenarios {1, 2, 3, 4} according to the agents constraints in the scenarios. In each scenario task, the size of training step is 20,000, and the size of replay buffer is 1000.

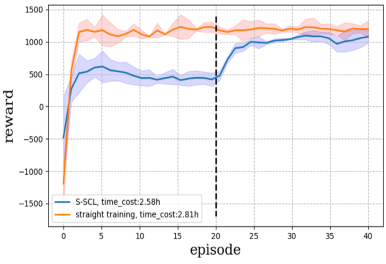


Fig. 10. Straight training and S-SCL in scenario 2.

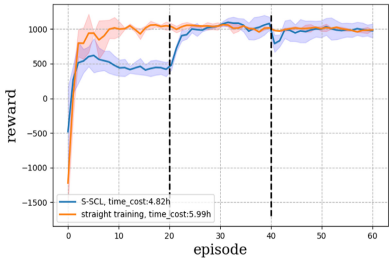


Fig. 11. Straight training and S-SCL in scenario 3.

First, the task curriculum includes scenario {1,2}. Figure 10 shows the comparison of two stages S-SCL and straight training reward in scenario 2. From the above-mentioned evaluation indicators, jumpstart and computation time of S-SCL are superior to those of straight training. Moreover, learning speed improvement and asymptotic improvement of S-SCL are worse than those of straight training. Figure 13 shows the win rate comparison of the above two methods. We can see that straight training is more effective. The reason is that the scene is relatively simple and the reward functions are relatively sparse. Therefore, the training step size of straight training is larger for scenario 2, and then more effective policy can be explored to obtain higher reward scores than pursuit policy trained in S-SCL. Thus, the method of straight training has more advantages in the reward and win rate indicators.

Next, we will verify the effect between two methods in scenario 3 as shown in Fig. 11. The jumpstart improvement and computation time indicates of S-SCL still have advantage than those of straight training. The learning speed of S-SCL is worse than that of straight training, while the asymptotic improvement is basically the same. The poor effect of learning speed improvement is due to the fact that the reward about interfering robots is not considered in the starting stage. The total number of agents is different, so the total score is not comparable. According to Fig. 11 and Fig. 13, the training time is reduced by 19.5% and the win rate is increased from 92.5% to 95%.

Figure 12 describes the advantage of S-SCL in scenario 4. The computation time of S-SCL is reduced by 13%, and the win rate is increased from 85.69% to 94.4% in Fig. 13. Therefore, the S-SCL method is superior to straight training. For the corresponding stages of scenario 3 and scenario 4, since the constraints of these two parts are negative reward feedback, there is a reward level decline for the curve in Figs. 11 and 12.

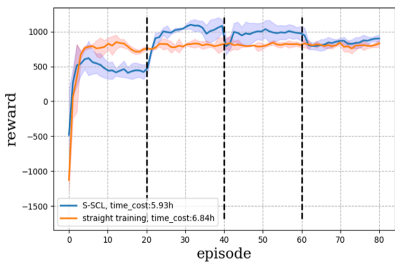


Fig. 12. Straight training and S-SCL in scenario 4.

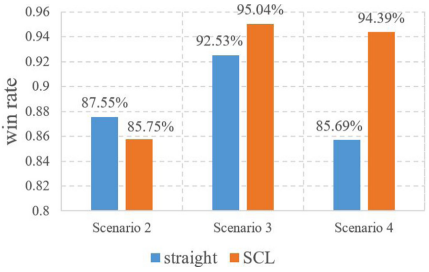


Fig. 13. Win rates between S-SCL and straight training methods in scenarios.

To sum up, under complex scenario conditions and non-sparse rewards, S-SCL can effectively reduce the computation time as well as improve training reward and win rates.

Asymmetric SCL. We observe that the results in the previous model can be used as the beneficial initial conditions for the next scenario. In this part, we explore the effect of different curriculum order on the experimental results. We designed the task curriculum as scenario 1, scenario IRAS (Interfering Robot Added into Scenario 1), scenario 3, and scenario 4. Based on the condition of global perception and communication in Subsect. 5.1, the IRAS introduces interfering robots, and each robot has a global observation perspective. Then, the constraint of the electromagnetic domain is introduced in scenario 3.

In scenario 1, the training step size is 20,000 and the replay buffer is 500, and the objective is to better train the collaborative pursuing ability of predator robots. In scenario IRAS, the replay buffer is 700, and the objective is to

further train the confrontation ability of both sides in the electromagnetic field under the collaborative pursuing behavior. In Fig. 14, we do not consider the learning speed improvement due to the inconsistent number of agents. From the aspects of jumpstart improvement and asymptotic improvement, we observe A-SCL to show a great improvement in both initial and final rewards, with a slight advantage in the computation time. The left-side results in Fig. 17 indicates that A-SCL increases the win rate of straight training method from 93% to 97%.

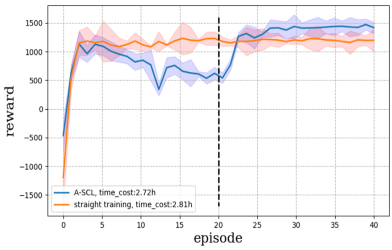


Fig. 14. Straight training and A-SCL in scenario 2.

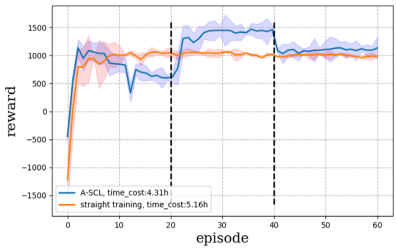


Fig. 15. Straight training and A-SCL in scenario 3.

Based on simulations in the previous part, the size of the replay buffer in scenario 3 is 900. Its purpose is to learn the strategy of maintaining communication cooperation in the situation of signal jamming. From the reward comparison of the three-stages A-SCL and straight training shown in Fig. 15, we observe that reward of A-SCL is slightly higher than those of straight training, and the computation time is reduced by 16.4%. Figure 17 indicates that A-SCL increases the win rate of straight training from 90% to 91%.

The size of the replay buffer in scenario 4 is 1100, and it aims to learn the obstacle avoidance function in the geographic domain in the electromagnetic domain condition. Figure 16 shows that the reward of A-SCL is higher than that of straight training, and the computation time is reduced by 11.4%. Figure 17 indicates that A-SCL increases the win rate of straight training from 87% to 97%.

In conclusion, we can design the size of the replay buffer manually, which will affect the weights of the trained agent models. The trained models are used as the initial condition for the subsequent scenario. Therefore, A-SCL has greater effect on the computation time, jumpstart improvement, and asymptotic improvement.

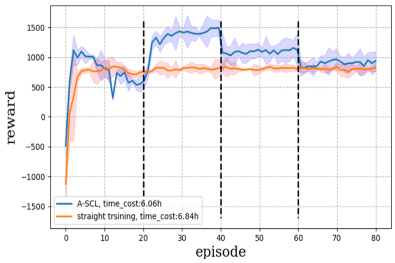


Fig. 16. Straight training and A-SCL in scenario 4.

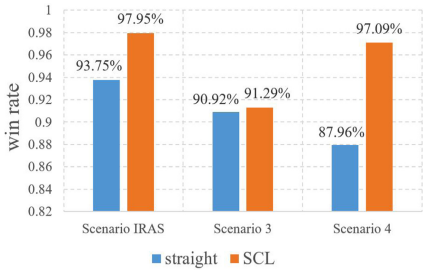


Fig. 17. Comparison of win rates from A-SCL and straight training.

7 Conclusion

In this paper, we considered a scenario where a team of predator robots collaboratively survey an area for prevention of invasion from opponent robots. To maximize the odds in a "cooperative–competitive" scenario, we adopted the cumulative reward as the performance metric. We modified the DRL algorithms for the scenarios for obtaining the optimal model of collaborative surround strategy under the constraint of maintaining communication quality and maximum sensor scope. We tackled the complex scenarios to further improve the SCL training method. Finally, the simulation results showed the effectiveness of the solution model in complex multi-domain problems and that the SCL method could improve efficiently and reduce the training time by 13%.

References

1. Bowling, M., Veloso, M.: Multiagent learning using a variable learning rate. Artif. Intell. **136**(2), 215–250 (2002)
2. Wu, Y., Ren, X., Zhou, H., Wang, Y., Yi, X.: A survey on multi-robot coordination in electromagnetic adversarial environment: challenges and techniques. IEEE Access **8**, 53484–53497 (2020)
3. Usunier, N., Synnaeve, G., Lin, Z., Chintala, S.: Episodic exploration for deep deterministic policies: an application to StarCraft micromanagement tasks. arXiv preprint arXiv:1609.02993 (2016)
4. Buşoniu, L., Babuška, R., De Schutter, B.: Multi-agent reinforcement learning: an overview. In: Srinivasan, D., Jain, L.C. (eds.) Innovations in Multi-Agent Systems and Applications - 1. Studies in Computational Intelligence, vol. 310. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14435-6_7
5. Li, Y.: Deep reinforcement learning: an overview. arXiv preprint arXiv:1701.07274 (2017)
6. Hernandez-Leal, P., Kartal, B., Taylor, M.E.: Is multiagent deep reinforcement learning the answer or the question? A brief survey. arXiv preprint arXiv:1810.05587 (2018)
7. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)

8. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O.P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Advances in Neural Information Processing Systems, pp. 6379–6390 (2017)
9. Foerster, J.N., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of AAAI Conference on Artificial Intelligence (2018)
10. Papoudakis, G., Christianos, F., Rahman, A., Albrecht, S.V.: Dealing with non-stationarity in multi-agent deep reinforcement learning. arXiv preprint arXiv:1906.04737 (2019)
11. Machado, M.C., Bellemare, M.G., Talvitie, E., Veness, J., Hausknecht, M., Bowling, M.: Revisiting the arcade learning environment: evaluation protocols and open problems for general agents. J. Artif. Intell. Res. **61**, 523–562 (2018)
12. Samvelyan, M., et al.: The StarCraft multi-agent challenge. In: Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems, pp. 2186–2188 (2019)
13. Bard, N., et al.: The Hanabi challenge: a new frontier for AI research. Artif. Intell. **280**, 103216 (2020)
14. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Machine Learning Proceedings 1994, pp. 157–163. Elsevier (1994)
15. Wu, Y., Zhang, B., Yi, X., Tang, Y.: Communication-motion planning for wireless relay-assisted multi-robot system. IEEE Wirel. Commun. Lett. **5**(6), 568–571 (2016)
16. Wu, Y., Zhang, B., Yang, S., Yi, X., Yang, X.: Energy-efficient joint communication-motion planning for relay-assisted wireless robot surveillance. In: Proceedings of IEEE Conference on Computer Communications, pp. 1–9. IEEE (2017)
17. Kshemkalyani, A., Ali, F.: Fast graph exploration by a mobile robot. In: Proceedings of International Conference on Artificial Intelligence and Knowledge Engineering (2018)
18. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. In: Proceedings of AAAI Conference on Artificial Intelligence (2018)
19. Narvekar, S., Sinapov, J., Leonetti, M., Stone, P.: Source task creation for curriculum learning. In: Proceedings of International Conference on Autonomous Agents & Multiagent Systems, pp. 566–574 (2016)
20. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of Annual International Conference on Machine Learning, pp. 41–48. ACM (2009)
21. Andreas, J., Klein, D., Levine, S.: Modular multitask reinforcement learning with policy sketches. In: Proceedings of International Conference on Machine Learning, pp. 166–175. JMLR. org (2017)
22. Wu, Y., Tian, Y.: Training agent for first-person shooter game with actor-critic curriculum learning. In: Proceedings of International Conference on Learning Representations (2016)
23. Wang, W., et al.: From few to more: large-scale dynamic multiagent curriculum learning. arXiv preprint arXiv:1909.02790 (2019)
24. Madden, M.G., Howley, T.: Transfer of experience between reinforcement learning environments with progressive difficulty. Artif. Intell. Rev. **21**(3–4), 375–398 (2004)
25. Lazaric, A.: Transfer in reinforcement learning: a framework and a survey. In: Wiering, M., van Otterlo, M. (eds.) Reinforcement Learning. Adaptation, Learning, and Optimization, vol. 12. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27645-3_5