DOI: 10.17587/mau.23.507-514

V. I. Petrenko, vipetrenko@ncfu.ru, **F. B. Tebueva**, ftebueva@ncfu.ru,
M. M. Gurchinsky, gurcmikhail@yandex.ru, **A. S. Pavlov,** anspavlov@ncfu.ru,
North-Caucasus Federal University, Stavropol, 355017, Russian Federation

Corresponding author: **Petrenko Vyacheslav I.,** Cand. of Tech. Sc., Head of the department of organization and technology of information security, North-Caucasus Federal University, Stavropol, 355017, Russian Federation, e-mail: vipetrenko@ncfu.ru

Accepted on July 11, 2022

Method of Multi-Agent Reinforcement Learning in Systems with a Variable Number of Agents

Abstract

Multi-agent reinforcement learning methods are one of the newest and actively developing areas of machine learning. Among the methods of multi-agent reinforcement learning, one of the most promising is the MADDPG method, the advantage of which is the high convergence of the learning process. The disadvantage of the MADDPG method is the need to ensure the equality of the number of agents N at the training stage and the number of agents K at the functioning stage. At the same time, target multi-agent systems (MAS), such as groups of UAVs or mobile ground robots, are systems with a variable number of agents, which does not allow the use of the MADDPG method in them. To solve this problem, the article proposes an improved MADDPG method for multi-agent reinforcement learning in systems with a variable number of agents. The improved MADDPG method is based on the hypothesis that to perform its functions, an agent needs information about the state of not all other MAS agents, but only a few nearest neighbors. Based on this hypothesis, a method of hybrid joint / independent learning of MAS with a variable number of agents is proposed, which involves training a small number of agents N to ensure the functioning of an arbitrary number of agents K, K> N. The experiments have shown that the improved MADDPG method provides an efficiency of MAS functioning com-parable to the original method with varying the number of K agents at the stage of functioning within wide limits.

Keywords: *multi-agent systems, machine learning, multi-agent reinforcement learning, collaborative learning, independent learning, variable number of agents*

For citation:

Petrenko V. I., Tebueva F. B., Gurchinsky M. M., Pavlov A. S. Method of Multi-Agent Reinforcement Learning in Systems with a Variable Number of Agents, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2022, vol. 23, no. 10, pp. 507—514.

DOI: 10.17587/mau.23.507-514

УДК 004.852

DOI: 10.17587/mau.23.507-514

В. И. Петренко, канд. техн. наук, зав. кафедрой организации и технологии защиты информации, vipetrenko@ncfu.ru,
Ф. Б. Тебуева, д-р физ.-мат. наук, зав. кафедрой компьютерной безопасности, ftebueva@ncfu.ru,
М. М. Гурчинский, аспирант, программист лаборатории робототехнических систем, gurcmikhail@yandex.ru,
А. С. Павлов, инженер-лаборант кафедры компьютерной безопасности, anspavlov@ncfu.ru,
Северо-Кавказский федеральный университет, Ставрополь, Россия

Метод мультиагентного обучения с подкреплением в системах с переменным числом агентов

Методы мультиагентного обучения с подкреплением являются одним из новейших и активно развивающихся направлений машинного обучения. Среди методов мультиагентного обучения с подкреплением одним из наиболее перспективных является метод MADDPG, достоинством которого является высокая сходимость процесса обучения. Недостатком метода MADDPG является необходимость обеспечения равенства числа агентов N на стадии обучения

и числа агентов K на стадии функционирования. В то же время целевые мультиагентные системы (МАС), такие как группы БПЛА или мобильных наземных роботов, являются системами с переменным числом агентов, что не позволяет применять в них метод MADDPG. Для решения данной проблемы в статье предложен усовершенствованный метод MADDPG для мультиагентного обучения с подкреплением в системах с переменным числом агентов. Усовершенствованный метод MADDPG базируется на гипотезе о том, что для выполнения своих функций агенту нужна информация о состоянии не всех прочих агентов МАС, а только нескольких ближайших соседей. На основе данной гипотезы предложен метод гибридного совместного/независимого обучения МАС с переменным числом агентов, который предполагает обучение некоторого небольшого числа агентов N для обеспечения функционирования произвольного числа агентов K, K > N. Проведенные эксперименты показали, что усовершенствованный метод MADDPG обеспечивает сопоставимую с оригинальным методом эффективность функционирования МАС при варьировании числа K агентов на стадии функционирования в широких пределах.

Ключевые слова: *мультиагентные системы, машинное обучение, мультиагентное обучение с подкреплением, совместное обучение, независимое обучение, переменное число агентов*

Introduction

With the development of reinforcement learning methods more and more com-plex problems fall into the area of interest of researchers. One of the newest and actively developing areas of reinforcement learning is multi-agent reinforcement learning MDRL multi-agent systems (MAS). The interest in the MAS is due to two rea-sons: 1) economic efficiency of using, instead of one complex agent, a set of technically simpler agents with comparable total productivity; 2) higher efficiency of decentralized problem-solving using MAS in comparison with similar centralized methods [1—3].

MAS find application in such areas [4] how production [5], transportation, smart homes, robotics [6—9], aviation, infrastructure facilities, medicine [10] and etc. An important problem in the development of MAS is the ambiguity and nontriviality of the synthesis of the policy of behavior of individual agents according to the specified target indicators of the MAS [11—13]. At the same time, single-agent reinforcement learning SDRL has established itself as a powerful and versatile tool for solving intellectual problems at a level comparable to that of a person [13—15]. Taken together, these factors determine the relevance of development single-agent deep reinforcement learning (SDRL) for use in MAS in the form of MDRL.

The two main challenges of MDRL are learning convergence and scalability [16]. The scalability problem also includes the problem of the possibility of dynamically changing the number of agents during the operation of the MAS. The problem of the possibility of dynamically changing the number of agents is of particular importance in the context of mobile MAS. Mobile MAS such as UAV groups [17] or ground mobile robots, in comparison with other types of MAS, are characterized by functioning in a non-deterministic environment with a relatively high probability of failure of individual agents, unstable bandwidth and topology of communication channels [18] and etc. Also, in mobile MAS, higher requirements for the reliability of control methods [19, 20]. Calculation error [21, 22] can damage and disable mobile MAS agents. These factors lead to the fact that mobile MAS in the general case can be considered as MAS with a variable number of agents.

To ensure the operation of the MAS with a variable number of agents, the article proposes an improved method of deep multi-agent learning with reinforcement based on the gradient of the deterministic policy MADDPG. The article is structured as follows. Section 2 reviews the existing methods and problems of MDRL, substantiates the use of the MADDPG method as a prototype, and describes its main features. Section 3 describes the proposed improved MADDPG method. Section 4 presents experimental results confirming the effectiveness of the proposed improved MADDPG method.

Target setting

The target set in this paper is to develop a scalable MDRL method based on the MADDPG. The developed method must be able to work with a variable number of agents. In this case, the computational complexity should grow no more than linearly in a wide range of the number of MAS agents.

Current state of the problem

The subject of this work is such a type of MOP as learn cooperation [16]. The current section examines various approaches and methods of teaching cooperation from the point of view of their effectiveness, mechanisms for overcoming the nonstationary of the environment in the process of learning and functioning, and the possibility of working with

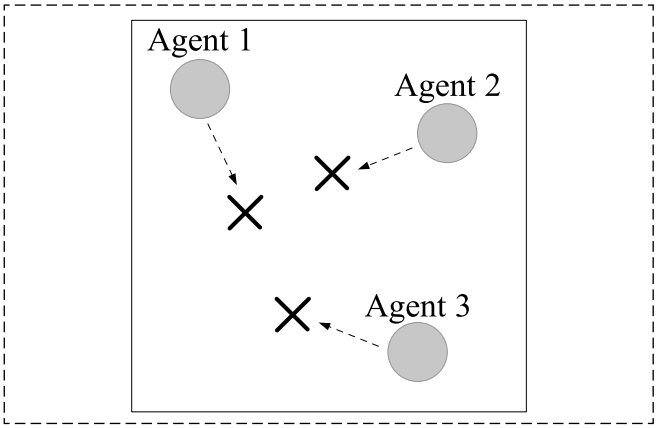

Fig. 1. An example of a spatially distributed problem [23]

a variable number of agents. This paper discusses MOS methods for cooperative MAS without explicit pre-organized or self-organizing communication, designed to solve spatially distributed problems. A generalized example of such a task is given in [23] under the name "Cooperative navigation", is shown on Fig. 1.

The problem includes an MAS of K agents (green circles) and an environment in which there is a set of K target objects (black crosses). The goal of MAS agents in this task is to occupy all target objects.

Let us introduce the following notation to describe the interaction between the MAS and the environment:

s — state of the environment, which also includes the state of agents $s \in S$, where S is set of possible states of the environment;

o_i — observation state of the environment available i-th agent, $o_i \in O_i$ where O_i is the set of possible observable states of the environment of the i-th agent;

a_i — action, undertaken by the i-th agent, $a_i \in A_i$ where A_i — set of possible actions of the i-th agent;

$r_i \in \mathrm{R}$ — reward, received by the i-th agent for performing an action a_i in a state of environment s, and the transition of the environment to the state s';

π_i — the policy of the i-th agent used by the i-th agent to transform the current observable state of the environment o_i in the probability distribution of the action to be taken a_i and parameterized by some vector of parameters θ_{π_i}:

$$\pi_i : O_i \times A_i \to [0;1], \tag{1}$$

μ_i — notation for deterministic policy π_i:

$$\mu_i : O_i \times A_i \to \{0;1\}, \tag{2}$$

Q_i — value function used by the i-th agent to estimate the expected value of the future cumulative reward when performing an action a_i in the observed state of the environment o_i, and parameterized by some vector of parameters θ_{Q_i}:

$$Q_i : O_i \times A_i \to \mathbb{R}. \tag{3}$$

According to the concept of reinforcement learning used by a policy agent π_i and the value function Q_i are approximations of some ideal policy π_i^* and/or ideal value function Q_i^*. In the process of training, the optimization of the parameter vectors takes place θ_{π_i} and θ_{Q_i} to improve the accuracy of the approximation. Deep reinforcement learning uses artificial neural networks as approximators (ANNs), and as vectors of optimized parameters θ_{π_i} and θ_{Q_i} the weights of the respective ANNs.

Methods MDRL can be divided according to the object of approximation into value-based methods and policy gradient methods, under the conditions of training on methods of joint training of agents and methods of independent training of agents. In value methods, the object of approximation is the value function Q_i, in gradient methods, the object of approximation is the policy π_i without/with value function Q_i. Value-base methods [24—31] presented mainly by methods based on the learning algorithm DQN [14] and its modifications DRQN [32]. The main disadvantages of value-based methods are the low learning efficiency compared to gradient methods. [33, 34] and focus on working with discrete action space A_i. An example of a gradient method for independent learning of agents is the method FTW [35]. The FTW method is an adaptation of the SDRL method based on the actor-learner structure IMPALA [36]. In article [23] it is shown that adapted gradient SDRL methods in multi-agent problems have weak convergence. This phenomenon is because the reward received by the i-th agent r_i depends not only on his actions a_i also from the actions of other agents $\{a_j \mid j \neq i\}$, therefore, the probability of obtaining the correct direction of the gradient with an increase in the number of agents decreases exponentially. To solve this problem at work [23] proposed a gradient method for collaborative learning of MADDPG agents (Multiagent Deep Deterministic Policy Gradient), DDPG-based (Deep Deterministic Policy Gradient) [37]. The DDPG method is based on the use of two ANNs. First ANNs − actor, used to approximate deterministic policy μ_i another ANNs. critic,

used to approximate the value function Q_i. In the MADDPG method, each agent uses its own ANNs pair actor and critic. At the entrance of the ANNs critic i-th agent approximating the value function Q_i, not only action is given a_i i-th agent, but also the actions of all other agents $\{a_j \mid j \neq i\}$. An important disadvantage of the MADDPG method is the need to ensure equality of the number of agents at the training stage. N and the number of agents at the stage of functioning K.

Based on the analysis of publications, the following conclusions can be drawn. The MDRL value methods are applicable only for problems with a discrete set of actions. Gradient MDRL methods with collaborative learning of agents are applicable to problems with a continuous set of actions and have good convergence, but they have poor scalability and fault tolerance. Gradient MDRL methods with independent learning of agents have good scalability and fault tolerance, however, they have problems with overcoming the nonstationary of the environment [16].

The listed features of gradient MDRL methods with joint and independent training of agents determine the relevance of the development of hybrid gradient MDRL methods with high convergence, scalability, and fault tolerance. In this paper, we propose such a hybrid method based on the MADDPG method and the concept of parameter separation. Details of the implementation of the parameter sharing concept in the enhanced MADDPG method are described in the next Section.

MADDPG method

In this work, the MADDPG method is taken as a basis for improvement in view of its following advantages [23]:

1) versatility of work in cooperative and competitive environments;

2) efficiency in comparison with other gradient methods of joint MDRL;

3) the choice of actions can be carried out from a continuous set of possible values.

The prototype of the multi-agent MADDPG method is the single-agent DDPG method. [37]. The essence of the DDPG method is as follows. Let there be a deterministic policy $\mu(o|\theta_\mu)$ making decisions by an agent to take an action a in the observed state of the environment o, parameterized by a parameter vector θ_μ. When describing the single-agent DDPG method, the subscript "i", indicating the agent number, not used in values. The DDPG method is based on the optimization of the parameter vector θ_μ with the aim of maximizing the expected total discounted remuneration J. New value of the vector of parameters θ_μ^{t+1} at each optimization iteration is determined by the formula:

$$\theta_\mu^{t+1} = \theta_\mu^t + \alpha\widehat{\nabla_{\theta_\mu^t} J}, \tag{4}$$

where θ_μ^t — the current value of the parameter vector θ_μ; α — optimization step size; $\nabla_{\theta_\mu^t} J$ — gradient J over the vector of parameters θ_μ^t; $\widehat{\nabla_{\theta_\mu^t} J}$ — estimated gradient value $\nabla_{\theta_\mu^t} J$.

According to the policy gradient theorem [38], provided that the medium is partially observable for the gradient $\nabla_{\theta_\mu^t} J$ the following relation is true:

$$\begin{aligned}\widehat{\nabla_{\theta_\mu^t} J} &\approx \mathbb{E}_{s\sim p^\mu}\left[\nabla_{\theta_\mu^t} Q_\mu(o,a \mid \theta_Q^t)\Big|_{o=o_t, a\sim\mu(o_t|\theta_\mu^t)}\right] = \\ &= \mathbb{E}_{s\sim p^\mu}\left[\nabla_a Q_\mu(o,a \mid \theta_Q^t)\Big|_{o=o_t, a\sim\mu(o_t|\theta_\mu^t)} \times \right. \\ &\left. \times \nabla_{\theta_\mu}\mu(s \mid \theta_\mu^t)\Big|_{s=s_t}\right],\end{aligned} \tag{5}$$

where $\mathbb{E}_{s_t\sim p^\mu}[x]$ — mathematical expectation of a quantity x provided that the state of the environment s_t has distribution p, policy-driven μ; $Q_\mu(o,a \mid \theta_Q^t)$ — value function Q, used by an agent at a point in time t when following a policy μ, taking as arguments the state of the environment o observed by the agent and the action taken by the agent a, parameterized by a parameter vector θ_Q^t; $a \sim \mu(o_t \mid \theta_\mu^t)$ means the agent chooses an action a according to the distribution returned by $\mu(o_t \mid \theta_\mu^t)$.

Ideal value function Q_μ^* with full observability of the environment ($o \equiv s$) is given by the Bellman equation:

$$Q_\mu^*(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1}\sim E^\mu}[r_t + \gamma Q_\mu^*(s_{t+1}, \mu(s_{t+1}))], \tag{6}$$

where $\mathbb{E}_{r_t, s_{t+1}\sim E^\mu}[x]$ — mathematical expectation of a quantity x subject to receiving the award r_t and the transition of the environment to some next state s_{t+1} according to the distribution returned by the math entity E when using the policy μ; E — a mathematical entity describing the functioning of an environment, usually described as a mathematical model or an algorithm for the dynamics of the environment; $\gamma \in [0;1]$ — a discounting parameter reflecting a

decrease in the importance of a subsequent reward r_{t+1} compared to the current award r_t.

In DDPG method for policy approximation μ and value functions Q_μ two ANNs are used, respectively named actor and critic. The weights of the ANNs actor act as a vector of parameters θ_μ policy μ, and the ANN weights critic act as a vector of parameters θ_Q value functions Q_μ. During training, optimization of the vector of parameters θ_μ is carried out according to the formula (4). Optimization of parameter vector values θ_Q is performed by a gradient machine learning method in order to minimize the following loss function:

$$L(\theta_Q) = \mathbb{E}_{s_t \sim p^\mu, a_t \sim \mu, r_t \sim E^\mu}[(Q_\mu(o_t, a_t \mid \theta_Q^t) - y_t)^2], \quad (7)$$

$$y_t = r(s_t, a_t) + \gamma Q^\mu(o_{t+1}, a_{t+1} \mid \theta_Q^t), \quad (8)$$

where y_t — expected total discounted remuneration.

The MADDPG method is a multi-agent extension of the DDPG method. For each i-th agent from N agents MAS uses two ANNs (Fig. 2, see the second side of the cover) for policy approximation μ_i and value functions Q_i — actor and critic, respectively. At the stages of learning and functioning i-th agent uses decision policy μ_i for transform the current observable state of the environment o_i in the action taken a_i. Value functions Q_i the i-th agent is used only at the training stage to assess the usefulness of the action taken by the i-th agent a_i.

In contrast to the single-agent variant of the value function $Q(o, a)$ the DDPG method, which takes as an argument the action a of one agent, a multi-agent value function $Q_i(o_i, a_i)$ method MADDPG takes as an argument the actions of all agents:

$$a = \{a_i \mid i = \overline{1, N}\}. \quad (9)$$

The following is a more rigorous mathematical description of the method, MADDPG. Consider a problem with N agents whose policies μ_i, $i = \overline{1, N}$ parameterized by parameter vectors θ_{μ_i}. Optimizing parameter vectors θ_{μ_i} using the gradient rise method, it is carried out according to the formula:

$$\theta_{\mu_i}^{t+1} = \theta_{\mu_i}^t + \alpha \widehat{\nabla_{\theta_{\mu_i}^t} J_i}, \quad (10)$$

where J_i — expected total discounted remuneration of the i-th agent.

Gradient $\widehat{\nabla_{\theta_{\mu_i}^t} J_i}$ calculated by the formula:

$$\widehat{\nabla_{\theta_{\mu_i}^t} J_i} \approx \mathbb{E}_{s \sim p^{\mu_i}} \left[\nabla_{a i} Q_{\mu_i}(o, a \mid \theta_{Q_i}^t)\Big|_{o=o_t, a_i \sim \mu_i(o_t \mid \theta_{\mu_i}^t)} \times \right.$$
$$\left. \times \nabla_{\theta_{\mu_i}} \mu_i(s \mid \theta_{\mu_i}^t)\Big|_{s=s_t} \right], \quad (11)$$

where $\theta_{Q_i}^t$ — vector of parameters of the value function Q_{μ_i} now t.

Q_{μ_i} updated to minimize the loss function $\mathcal{L}(\theta_{Q_i})$:

$$L(\theta_{Q_i}) = $$
$$= \mathbb{E}_{s_t \sim p^{\mu_i}, a_t \sim \mu_i, r_t \sim E}[(Q_{\mu_i}(o_t, a_t \mid \theta_{Q_i}^t) - y_t)^2], \quad (12)$$

$$y_t = r(s_t, a_t) + \gamma Q_{\mu_i}(o_{t+1}, a_{t+1} \mid \theta_{Q_i}^t). \quad (13)$$

In the article [23] the effectiveness of the MADDPG method was investigated when working with a fixed number of agents, the same for the training stage and for the execution stage. As the authors note, one of the further directions in the development of the method is its development for working with a variable number of agents.

Advanced MADDPG Method

The advanced MADDPG method differs from the original method in the following modifications.

1. At the training stage, N agents are used, at the functioning stage, an arbitrary number of $K > N$ agents.

2. At the stage of functioning, a "scope" is introduced for agents. The state of the environment o_i observed by the i-th agent includes the state of only those agents that fall into this scope. Let's designate the field of visibility, determined by objective reasons, such as the range of the communication channel or the range of visual observation, as the natural field of visibility. If more than $N - 1$ agents, the method uses an artificial scope, which includes $N - 1$ nearest agents (Fig. 3, *a*, see the second side of the cover). If the natural scope falls $M < N - 1$ other agents as states $(N - 1) - M$ of unobserved agents, their last observed state is used (Fig. 3, *b*, see the second side of the cover).

3. According to the concept of parameter sharing, the same decision-making policies are used for all agents. $\mu_i = ... = \mu_N = \mu$ and the value function $Q_1 = ... Q_N = Q$ (Fig. 4, see the second side of the

cover). At the training stage, one ANN actor and one ANN critic are used for all agents. At the stage of functioning, all agents use identical copies of the actor ANN.

Expressions (10)—(13), taking into account the listed features, take the following form. Common policy for all agents μ parameterized by the parameter vector θ_μ. Parameter vector optimization θ_μ uing the gradient rise method, it is carried out according to the formula:

$$\theta_\mu^{t+1} = \theta_\mu^t + \alpha\widehat{\nabla_{\theta_\mu^t} J_i}, \tag{14}$$

where J_i — expected total discounted remuneration of the i-th agent.

Gradient $\widehat{\nabla_{\theta_\mu^t} J_i}$, calculated by the formula:

$$\widehat{\nabla_{\theta_\mu^t} J_i} \approx \mathbb{E}_{s\sim p^\mu}\left[\nabla_{ai} Q_\mu(o, a \mid \theta_Q^t)\Big|_{o=o_t, a_i\sim\mu_i(o_t|\theta_{\mu_i}^t)} \times \nabla_{\theta_{\mu_i}}\mu_i(s \mid \theta_{\mu_i}^t)\Big|_{s=s_t}\right], \tag{15}$$

where θ_Q^t — vector of parameters of the value function Q_μ at the moment t.

Q_μ updated to minimize the loss function $\mathcal{L}(\theta_Q)$:

$$L(\theta_Q) = \mathbb{E}_{s_t\sim p^\mu, a_t\sim\mu, r_t\sim E}[(Q_\mu(o, a \mid \theta_Q^t) - y_t)^2], \tag{16}$$

$$y_t = r(s_t, a_t) + \gamma Q_\mu(o_{t+1}, a_{t+1} \mid \theta_Q^t). \tag{17}$$

The modifications of the MADDPG method proposed in the work lead to the following results:

1. Computational complexity of each training step with an equal number of agents being trained N remains the same. The number of ANN actor and critic decreases by N times, while the number of acts of optimization of the weights of these ANNs increases by the same number of times at each learning step.

2. The introduction of the scope allows us to reduce the task of ensuring the functioning K agents to the task of ensuring the functioning N agents. As follows from the results of experimental studies in the next section, it is enough to conduct training $N < K$ agents, in contrast to the original method, where $N = K$.

3. Because $N < K$, in the improved MADDPG method, it will be possible to significantly reduce the computational complexity of MAS training to ensure the functioning K agents. Consider the factors affecting the computational complexity of one step of training when switching from training N_1 agents for training N_2 agents in the MADDPG method with $N_2 > N_1$:

1) the number of ANN inputs actor and critic, which are responsible for information about other agents, increases by N_2/N_1 times, which leads to a quadratic increase in the number of connections in the ANN actor and critic and a corresponding increase in the number of necessary computational operations;

2) the input data space increases proportionally N_2/N_1, which may require an increase in the number of perceptrons in individual layers of the ANN and lead to an additional abrupt increase in the number of computational operations;

3) the number of ANNs actor and critic increases by N_2/N_1 times, which leads to a linear increase in the required memory for storing the ANN weights actor and critic.

Taken together, these factors lead to a non-linear increase in the computational complexity of each training step.

4. In the analogous method, in the learning process, agents "get used" to each other's behavior, since the input data is an ordered tuple, where information about the state of the i-th agent is in the corresponding i-th position of the input data. By using identical copies of the ANN actor, the impersonality of agents is achieved — one of the prerequisites for scalability.

Results

The results of an experimental comparison of the original and improved MADDPG method are shown in Fig. 5—7 (see the second side of the cover). Fig. 5 shows graphs of the frequency of training steps execution on the same hardware for the original and improved MADDPG method with the number of agents being trained. N = {3;5}. The jump in the frequency of training steps at the initial stage is explained by the accumulation of data in the retry buffer and the absence of an optimization operation for the ANN weights. The graphs show approximately equal frequency of learning steps for the original and improved method.

Fig. 6 (see the second side of the cover) shows graphs of learning curves — the dependence of the total reward received by the MAS during an episode on the number of learning steps for the number of agents being trained N = {3;5}. As it follows from Fig. 6, the learning rate for the improved and original method is approximately equal.

Fig. 7 (see the second side of the cover) shows the average values of the total reward received by the trained MAS per episode for the original and improved MADDPG methods for various values of the number of K agents at the stage of functioning. For the original method, the results were obtained for $N \equiv K$. For the improved method, the results are given for various N.

As follows from the results, the efficiency of the functioning of the MAS trained using the improved method is comparable to the efficiency of the MAS trained using the original MADDPG method. In this case, using the improved method, it suffices to train the number of agents N less than the number of agents K at the stage of functioning for different values of K. For example, an MAS trained using the improved method for $N = 3$ for $K = \{5;7\}$ demonstrates the same efficiency as the original method, which requires strict compliance $N \equiv K = 5$ or $N \equiv K = 7$.

Conclusion

In this paper, we propose an improved multi-agent reinforcement learning method based on the MADDPG deterministic policy gradient. The improved MADDPG method is based on the use of the concept of shared parameters and the introduction of artificial scoping for agents. The obtained results of experimental studies have confirmed the following theoretical expectations:

1) with an equal number of trained agents N, the computational complexity of the improved and original MADDPG methods are the same;

2) the efficiency of training using the improved method of N agents for an MAS of K agents for $K > N$ comparable to the learning efficiency with the original method of K agents.

The results obtained confirm the possibility of the improved MADDPG method working with a variable number of agents. In the future, the solutions proposed in this work can also be used to reduce the computational complexity of the original MADDPG method by reducing the number of trained agents N for a given number of K MAS agents.

References

1. **Kovács G., Yussupova N., Rizvanov D.** Resource management simulation using multi-agent approach and semantic constraints, *Pollack Period*, 2017, vol. 12, no. 1, pp. 45—58.

2. **Darintsev O., Migranov A.** Task Distribution Module for a Team of Robots Based on Genetic Algorithms: Synthesis Methodology and Testing, *Proceedings of the 21st International Conference; Complex Systems: Control and Modeling Problems, CSCMP 2019*, 2019, pp. 296—300.

3. **Darintsev O. V.** et al. Methods of a heterogeneous multi-agent robotic system group control, *Procedia Computer Science*, 2019, vol. 150, pp. 687—694.

4. **Wang L., Törngren M., Onori M.** Current status and advancement of cyber-physical systems in manufacturing, *Journal of Manufacturing Systems*, 2015, vol. 37, pp. 517—527.

5. **Munasypov R. A., Masalimov K. A.** Neural network models for diagnostics of complex technical objects state by example of electrochemical treatment process, *Proceedings of the 2nd International Ural Conference on Measurements*, UralCon 2017, 2017, pp. 156—160.

6. **Bonilla Venegas F. V., Moya M. J., Litvin A., Lukyanov E., Marín Pillajo L. E.** Modeling and Simulation of the Robot Mitsubishi RV-2JA controlled by electromyographic signals, *Enfoque UTE*, vol. 9 (2), pp. 208—222.

7. **Vokhmintcev A. V., Melnikov A. V., Mironov K. V., Burlutsky V. V.** Reconstruction of Three-Dimensional Maps Based on Closed-Form Solutions of the Variational Problem of Multisensor Data Registration, *Reports of the Academy of Sciences,* 2019, vol. 484, no. 6, pp. 672—677.

8. **Bogdanov A., Dudorov E., Permyakov A., Pronin A., Kutlubaev I.** Control system of a manipulator of the anthropomorphic robot FEDOR, *Proceedings of the International Conference on Developments in eSystems Engineering, DeSE*, 2019, pp. 449—454.

9. **Petrenko V., Tebueva F., Antonov V., Untewsky N., Gurchinsky M.** Energy-Efficient Path Planning: Designed Software Implementation, *Proceedings of the 21st International Workshop on Computer Science and Information Technologies (CSIT 2019),* 2019, vol. 3, pp. 112—118.

10. **Bogdanov M., Nasyrov D., Dumchikova I., Samigullin A.** Processing of Biomedical Data with Machine Learning, *Proceedings of the 21st International Workshop on Computer Science and Information Technologies (CSIT 2019),* 2019, vol. 3, pp. 6—16.

11. **Petrenko V., Tebueva F., Gurchinsky M., Antonov V.** A Robotic Complex Control Method Based on Deep Reinforcement Learning of Recurrent Neural Networks for Automatic Harvesting of Greenhouse Crops, *Proceedings of the 8th scientific Conference on Information Technologies For Intelligent Decision Making Support (ITIDS 2020),* 2020, vol. 174, pp. 340—346.

12. **Petrenko V., Tebueva F., Antonov V., Gurchinsky M., Ryabtsev S., Burianov A.** Cooperative Motion Planning Method for Two Anthropomorphic Manipulators, *Proceedings of the 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019),* 2019, vol. 166, pp. 146—151.

13. **Petrenko V., Tebueva F., Pavlov A., Antonov V., Kochanov M.** Path Planning Method in the Formation of the Configuration of a Multifunctional Modular Robot Using a Swarm Control Strategy, *Proceedings of the 7th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2019),* 2019, vol. 166, pp. 165—170.

14. **Mnih V.** et al. Human-level control through deep reinforcement learning, *Nature*, 2015, vol. 518, pp. 529—533.

15. **Petrenko V., Tebueva F., Pavlov A., Svistunov N.** Machine Learning Algorithm for Anthropomorphic Manipulator Control System, *Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020),* 2020, vol. 174, pp. 353—358.

16. **Hernandez-Leal P., Kartal B., Taylor M. E.** A survey and critique of multiagent deep reinforcement learning, *Autonomous Agents and Multi-Agent Systems*, 2019, vol. 33, pp. 750—797.

17. **Pshikhopov V., Medvedev M., Medvedeva T.** Terminal Motion Control of Multicopter Group, *Proceedings of the 4th International Conference on Control and Robotics Engineering, ICCRE 2019*, 2019, pp. 1—6.

18. **Wang H., Zhao H., Ma D., Wei J.** Cyber Physical System Framework for UAV Communications, *Electrical Engineering and Systems Science*, 2020, pp. 1—41.

19. **Yusupova N., Rizvanov D., Andrushko D.** Cyber-Physical Systems and Reliability Issues, *Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020),* 2020, vol. 174, pp. 133—137.

20. **Fabarisov T., Yusupova N., Ding K., Morozov A., Janschek K.** Model-based stochastic error propagation analysis for cyber-physical systems, *Acta Polytechnica Hungarica,* 2020, vol. 17, no. 8, pp. 15—28.

21. **Valiev E., Yusupova N., Morozov A., Janschek K., Beyer M.** Evaluation of the Impact of Random Computing Hardware Faults on the Performance of Convolutional Neural Networks, *Proceedings of the 8th Scientific Conference on Information Technologies for Intelligent Decision Making Support (ITIDS 2020),* 2020, vol. 174, pp. 307—312.

22. **Beyer M., Morozov A., Ding K., Ding S., and Janschek K.** Quantification of the impact of random hardware faults on safety-critical ai applications: Cnn-based traffic sign recognition case study, *Proceedings — 2019 IEEE 30th International Symposium on Software Reliability Engineering Workshops, ISSREW 2019,* 2019, pp. 118—119.

23. **Lowe R., Wu Y., Tamar A., Harb J., Abbeel P., Mordatch I.** Multi-agent actor-critic for mixed cooperative-competitive environments, *Advances in Neural Information Processing Systems*, 2017, vol. 2017-December, pp. 1—12.

24. **Foerster J., Nardelli N., Farquhar G., Afouras T., Torr P., Kohli P., Shimon Whiteson S.** Stabilising experience replay for deep multi-agent reinforcement learning, *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, 2017, vol. 70, pp. 1146—1155.

25. **Gupta J. K., Egorov M., Kochenderfer M.** Cooperative Multi-agent Control Using Deep Reinforcement Learning, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 2017, vol. 10642 LNAI, pp. 1—9.

26. **Bloembergen D., Kaisers M., Tuyls K.** Lenient frequency adjusted Q-learning, *Belgian/Netherlands Artificial Intelligence Conference*, 2010, pp. 19—26.

27. **Omidshafiei S., Pazis J., Amato C., How J. P., Vian J.** Deep decentralized multi-task multi-agent reinforcement learning under partial observability, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, 2017, vol. 6, pp. 4108—4122.

28. **Zheng Y., Jianye Hao J., Zhang Z.** Weighted double deep multiagent reinforcement learning in stochastic cooperative environments, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 2018, vol. 11013 LNAI, pp. 1—8.

29. **Hong Z. W., Shih-Yang Su S. Y., Shann T. Y., Chang Y. H., Lee C. Y.** A deep policy inference Q-network for multi-agent systems, *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS, 2018, vol. 2, pp. 1388—1396.

30. **Palmer G., Tuyls K., Bloembergen D., Savani R.** Lenient multi-agent deep reinforcement learning, *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2018, vol. 1, pp. 1—9.

31. **Matignon L., Laurent G. J., Le Fort-Piat N.** Hysteretic Q-Learning: An algorithm for decentralized reinforcement learning in cooperative multi-agent teams, *IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 1—7.

32. **Hausknecht M., Stone P.** Deep recurrent q-learning for partially observable MDPs, *AAAI Fall Symposium — Technical Report*, 2015, pp. 29—37.

33. **Matignon L., Laurent G. J., Le Fort-Piat N.** Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems, *Knowledge Engineering Review*, 2012, vol. 27, no. 1. pp. 1—32.

34. **Tan M.** Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, *Machine Learning Proceedings 1993*, 1993, pp. 1—8.

35. **Jaderberg M.** et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning, *Science,* 2019, vol. 364, no. 6443, pp. 859—865.

36. **Espeholt L.** et al. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018, vol. 4, pp. 1—10.

37. **Lillicrap T. P.** et al. Continuous control with deep reinforcement learning, *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016 — Conference Track Proceedings*, 2016, pp. 1—14.

38. **Foerster J. N., Assael Y. M., Nando de Freitas N., Whiteson S.** Learning to communicate with deep multi-agent reinforcement learning, *Advances in Neural Information Processing Systems*, 2016, pp. 1—9.

39. **Silver D., Lever G., Heess N., Degris T., Wierstra D., Riedmiller M.** Deterministic policy gradient algorithms, *Proceedings of the 31st International Conference on Machine Learning, ICML 2014,* 2014, vol. 32, pp. 387—395.