Promoting Training of Multi-Agent Systems

Petro Kravets[1[0000-0001-8569-423X]], Vasyl Lytvyn[2[0000-0002-9676-0180]], Victoria Vysotska[3[0000-0001-6417-3689]], Yevhen Burov[4[0000-0001-8653-1520]]

Lviv Polytechnic National University, Lviv, Ukraine

Petro.O.Kravets@lpnu.ua[1],Vasyl.V.Lytvyn@lpnu.ua[2], Victoria.A.Vysotska@lpnu.ua[3], Yevhen.V.Burov@lpnu.ua[4]

Abstract. The problem of incentive training of multi-agent systems in the game formulation for collective decision making under uncertainty is considered. Methods of incentive training do not require a mathematical model of the environment and enable decision making directly in the training process. Markov model of stochastic game is constructed and the criteria for its solution are formulated. An iterative Q-method for solving a stochastic game based on the numerical identification of a characteristic function of a dynamic system in space of state-action is described. Players' current gains are determined by the method of randomization of payment Q-matrix elements. Mixed player strategies are calculated using the Boltzmann method. Pure strategies are determined on the basis of discrete random distributions given by mixed player strategies. The algorithm for stochastic game solving is developed and results of computer implementation of game Q-method are analyzed.

Keywords – Multi-Agent System, Stochastic Game, Promotional Training, Q-method

1 Introduction

The functioning of most modern information systems (IS) is based on rigidly programmed algorithms. Unforeseen environmental influences in such systems may impair the stability of operating modes, which can lead to various types of emergency situations. To prevent critical states, distributed IS software must consist of interoperable standalone modules, be intelligent, flexible, and capable of independently monitoring environmental changes and making timely and appropriate decisions. Otherwise, such systems should be built on the principles of an agent-oriented methodology [1 – 10]. An IS agent is a standalone software module with elements of artificial intelligence, capable of making decisions on its own, interacting with the environment, other agents, and people as they accomplish the task. IS agents interact within the computer network. A population of computer network agents who solve a common problem is called a multi-agent system (MAS).

The operation of the MAS is usually carried out in the context of a priori uncertainty about the state of the decision-making environment and the actions of other

agents. In this regard, agent's behavior strategies must be adaptive at the expense of agents' ability to learn [11]. Among the methods of learning under uncertainty, incentive-based methods have gained practical appeal [12, 13] because they do not require a mathematical model of the environment and provide decision-making power directly in the learning process. The mechanisms of reflexive behavior of living organisms with developed nervous system are the basis of the stimulating training. An effective method of incentive learning is Markov Q-learning [14], which performs numerical identification of the characteristic function of a dynamic system in state-space. As a characteristic function is usually the function of the total expected reward agent.

Compared to single agent systems, the structure, operation, and research of multi-agent Q-learning methods are much more complicated [15]. Due to the collective interaction of agents, the stationary environment is transformed into a non-stationary class. The change in the state of the environment and the value of the benefits of each agent depend on the actions of the other agents. Generally, in an MAS, an agent cannot achieve a maximum gain equal to that of a single agent system. The optimal payoffs of agents must be balanced and meet the criteria of benefit, fairness, balance. Thus, instead of the criterion of scalar maximization of the benefits of a single-agent system, the criteria of vector maximization of MAS winnings are introduced, for example, Nash equilibrium, Pareto optimality or other [16].

Provided the use of methods of Q-learning of MAS, an iterative construction of a system of characteristic Q-functions in the state-action space takes place, and the increment of the elements of these functions is carried out in the direction of achieving their collective equilibrium. To build the MAS, it is necessary to carry out preliminary studies on the basis of adequate mathematical models that will allow to study the dynamics of the system under uncertainty, to build strategies for the behavior of agents that provide optimal technical and economic parameters of the system functioning. Given the peculiarities of the subject area, namely, multi-agency, uncertainty of decision-making environment, antagonism or competitiveness of goals, communicativeness, coordination of actions, adaptability of agent behavior strategies, we use the mathematical apparatus of stochastic game theory [17 – 29]. Solving a stochastic game is to find the strategies of agents that maximize their winnings so as to ensure a certain collective balance of interests for all players. The search for optimal strategies for players in uncertainty will be performed on the basis of the promotional training method.

The purpose of the work is to construct an iterative method of incentive learning to solve the stochastic MAS game in uncertainty. To achieve this goal, it is necessary to develop a model of multi-agent stochastic game, to determine the criteria of collective equilibrium, the method and algorithm for solving the game problem.

2 The Mathematical Model of Stochastic Game

The stochastic game is determined by the tuple:

$$(S, p, A_i, r^i \mid i \in I),\ I = \{1, 2, ..., L\},$$

where $S = \{s_1, ..., s_M\}$ is set of all states of the environment, $p : S \times A \to \Delta(S)$ is system state change function defined in the space of probability distributions $\Delta(S)$ on the plural S, $A_i = (a_i(1), ..., a_i(N_i))$ is multiple actions or pure strategies i-agent, $A = \underset{i \in I}{\times} A_i$ is set of combined agent actions, $r^i : S \times A \to R$ is reward function i-agent, I is multiple agents, L is number of agents, M is number of states, N_i is number of strategies i-agent.

In the general case of multiple actions $A_i = A_i(s)\ \ \forall i \in I$ and combined actions $A = A(s)$ may depend on the state of the environment $s \in S$.

We adopt a Markov model [30, 31] of the dynamics of states of a system in which the probability of change of states p depends only on the current state of the environment and the current actions of the agents:

$$p(s_{t+1} = s' \mid (s_\tau, b_\tau), \tau = 0, 1, 2, ..., t) = p(s_{t+1} = s' \mid s_t, b_t),$$

where $b_t \in A$ is combined action at time t.

At each point in time, the environment is in one of the states $s \in S$ and agents choose actions independently $a_i \in A_i$. After the implementation of the combined option $a = (a_1, ..., a_L) \in A$ agents get random winnings r^i (otherwise is incentives or reinforcements), and the environment changes its state according to the probability distribution $p(s, a)$ with values per segment $[0, 1]$:

$$\sum_{s' \in S} p(s' \mid s, a) = 1.$$

The agent implements actions based on a mixed strategy $\pi_i : S \to A_i$, which determines the likelihood of action $a_i \in A_i$ in every state of the environment $\forall s \in S$.

Distribution $\pi_i \in \Pi_i$ takes the value on a unit simplex [11]

$$\Pi_i = = \left\{ \pi \left| \sum_{a_i \in A_i} \pi(s, a_i) = 1, \pi(s, a_i) \geq 0 \right. \right\}.$$

If $\pi_i(s, a_i) \in \{0, 1\}$, then the agent determines the choices of solutions. Let the total payoff of each agent be determined by the function of discounted total payoffs:

$$\Upsilon_i = \sum_{t=0}^{\infty} \gamma^t r_t^i\ , \qquad (1)$$

where $\gamma \in (0, 1]$ is discount option.

The goal i-agent is to maximize function (1) by formulating an effective strategy π_i:

$$V_\pi^i(s) = E_\pi\left[\Upsilon_i \mid s_0 = s\right] \to \max_{\pi_i},\ \forall i \in I, \tag{2}$$

where $\pi = (\pi_1, ..., \pi_L)$; E is symbol of mathematical expectation.

Stochastic game resolution is about defining agent behavior strategies π_i^* ($\forall i \in I$), which ensure fulfilment of one of the conditions of collective optimality, for example:

1) Nash equilibrium: $V^i(s, \pi_1^*, \pi_2^*, ..., \pi_L^*) \geq V^i(s, \pi_1^*, \pi_2^*, ..., \pi_{i-1}^*, \pi_i, \pi_{i+1}^*, ..., \pi_L^*)$;

2) Pareto optimality: $V^i(s, \pi^*) \geq V^i(s, \pi)$.

3 Learning of Stochastic Game

Calculation $V_\pi^i(s)$ can be performed in a recursive form known in the literature as the Bellman equation [12 – 14]. Given (1), we obtain after simple transformations:

$$\begin{aligned} V_\pi(s \mid s_t = s) &= E(r_t) + \gamma \sum_{k=0}^{\infty} \gamma^k E(r_{t+k+1}) = E(r_t) + \gamma V_\pi(s_{t+1}) = \\ &= r(s, \pi(s)) + \gamma \sum_{s' \in S} p(s' \mid s, \pi(s)) V_\pi(s') \end{aligned} \tag{3}$$

where s' is probable future states of the system.

The agent's goal is to find a strategy π^* that maximizes function (2) for all states of the environment:

$$\forall \pi\ \forall s \in S \qquad V^{\pi^*}(s) \geq V^\pi(s).$$

Since the choice of options is made by chance, it is for the purpose of comparing the effectiveness of actions when the system is in a state $s \in S$, current gains are useful to obtain from (2). For this purpose is specially built Q is average payoff feature that determines the cost of the action – the total payoff of the agent in the state s chose action a:

$$Q_\pi(s, a) = E_\pi\left[R \middle| s_0 = s, a_0 = a\right]. \tag{4}$$

Expression (4) defines a tabular function of the values of the action options a in the states s.

Similarly to (3) we obtain:

$$Q_\pi(s, a) = r(s, a) + \gamma \sum_{s' \in S} p(s' \mid s, a) V_\pi(s'). \tag{5}$$

Adherence to the Bellman principle of optimality (5) ensures the optimal gain of the agent from the current state achieved $s \in S$ at all future times. Applying this principle to all states ensures a global optimal solution.

For optimal function selection strategies π^* for each state $s \in S$ we will get:

$$V_{\pi^*}(s) = \max_{a \in A} \left[r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V_{\pi^*}(s') \right]. \quad (6)$$

From (6) one can obtain the optimal function of choosing strategies

$$\pi^*(s) = \arg\max_{a \in A} Q_{\pi^*}(s,a). \quad (7)$$

Optimization (7) can be performed by dynamic programming methods [30].

By analogy to single agent training, we define the payoff matrix i-players with current and future winnings in the direction of movement to the optimal collective state in space $S \times A$:

$$Q_*^i(s,a_1,...,a_L) = r^i(s,a_1,...,a_L) + \gamma \sum_{s' \in S} p(s' \mid s,a_1,...,a_L) \cdot V^i(s',\pi_1^*,...,\pi_L^*),$$

where $Q_*^i(s,a_1,...,a_L)$ is total discounted gain i-player provided the players select the action $(a_1,...,a_L)$ in the state s according to the optimal strategy of the game $\pi^* = (\pi_1^*,...,\pi_L^*)$.

In the conditions of a priori uncertainty of the transition probabilities between the states of the system $p(s,a_1,...,a_L)$ and winnings features $r^i(s,a_1,...,a_L)$ an iterative method is used to calculate the elements of the payoff matrix Q-learning [31]:

$$Q_{t+1}^i(s,a_1,...,a_L) = (1-\alpha_t) Q_t^i(s,a_1,...,a_L) + \alpha_t [r_t^i + \gamma V_t^i(s_{t+1})] \quad (8)$$

where $\alpha_t \in (0, 1)$ is training option; $V_t^i(s_{t+1})$ is the operator of the cost of the system state in the direction of the optimal collective solution.

The type of the operator $V_t^i(s_{t+1})$ is determined by the condition of collective equilibrium, for example:

$V_t^i(s_{t+1}) = MM\left(Q_t(s_{t+1})\right)$ is maximin equilibrium;
$V_t^i(s_{t+1}) = NE\left(Q_t^i(s_{t+1})\right)$ is Nash equilibrium;
$V_t^i(s_{t+1}) = BR\left(Q_t^i(s_{t+1})\right)$ is best agent response;
$V_t^i(s_{t+1}) = CE\left(Q_t^i(s_{t+1})\right)$ is correlated equilibrium;
$V_t^i(s_{t+1}) = PE\left(Q_t^i(s_{t+1})\right)$ is Pareto optimality.

The above list may be supplemented by other already known and new equilibrium states that will determine the target aspect of the functioning of a distributed dynamic system. Method (8) can be applied to decipher a single agent game with nature as a partial case *N*-agent stochastic game if $I = \{i\}, |I| = 1, A = A_i$, when $V_t^i(s_{t+1}) = \max_{b \in A_i} Q_t^i(s', b)$, where $s' = s_{t+1}$.

The Maximin Equilibrium (MM) takes place in the game of two agents with zero sum of their payoff functions:

$$V^1(s_{t+1}) = \max_{\pi_1 \in \Pi_1} \min_{a_2 \in A_2} \sum_{a_1 \in A_1} \pi_1(s', a_1) Q_t(s', a_1, a_2) = -V^2(s_{t+1}) \ .$$

Nash Equilibrium (NE) is determined by the independent distribution of strategies by players who choose their own strategies independently of the choice of other agents. In a Nash equilibrium situation in mixed strategies $\pi^{NE}(s) = \left(\pi_1^{NE}(s), ..., \pi_L^{NE}(s)\right)$ it is not profitable for each agent to deviate from its own optimal strategy $\pi_i^{NE}(s)$, if other agents stick to the equilibrium point [31]:

$$\sum_{a \in A} Q_t^i(s', a) \pi_i^{NE}(a_i) \prod_{j \neq i}^{L} \pi_j^{NE}(s', a_j) \geq \sum_{a \in A} Q_t^i(s', a) \tilde{\pi}_i(a_i) \prod_{j \neq i}^{L} \pi_j^{NE}(s', a_j), \qquad (9)$$

where $a = (a_1, ..., a_L)$; $\pi_i^{NE}, \tilde{\pi}^i \in \Pi_i$.

Method (8) ensures that condition (9) is satisfied when the current value of the system state value operator is determined at point π^{NE} Nash equilibrium:

$$NE\left(Q_t^i(s_{t+1})\right) = \sum_{a \in A} Q_t^i(s', a) \prod_{j=1}^{L} \pi_j^{NE}(s', a_j) \ .$$

The set of NE equilibrium points in mixed strategies is a convex compact and can be calculated by linear programming methods (for bi-matrix games) or by solving a system of polylinear equations that determine the complementary rigidity condition:

$$\sum_{a_{-i} \in A_{-i}} Q_t^i(s, a_{-i}, a_i) \prod_{j \neq i}^{L} \pi_j(s, a_j) = \sum_{a \in A} Q_t^i(s, a) \prod_{j=1}^{L} \pi_j(s, a_j) \ , \ \forall i \in I \ , \ \forall a_i \in A_i \ , \ \forall s \in S \ ,$$

$$\pi_i(s, a_i) > 0 \ , \ \sum_{a_i \in A_i} \pi_i(s, a_i) = 1 \ .$$

Unlike the Nash equilibrium, the Best Response (BR) method generates an optimal agent strategy in response to the actions of all other agents. The corresponding system state value operator in method (8) has the form [32]:

$$BR\left(Q_t^i(s_{t+1})\right) = \max_{\pi_i}\left(\sum_{a\in A} Q_t^i(s',a)\prod_{j=1}^{L}\pi_j(s',a_j)\right).$$

Correlated Equilibrium (CE) generalizes the Nash equilibrium by allowing players' strategies to depend. For this purpose, there is an arbitrator in the collective decision-making system, which according to the generalized distribution $\sigma \in \Delta(A)$ ($\sum_{a\in A}\sigma(a) = 1$, $A = \underset{i=1}{\overset{L}{\times}} A_i$) recommends that players choose to take actions that form a combined option $a = (a_1,...,a_L)$. Player with a number i only receives component information combined option $a \in A$. This signal is perceived i-player as an optional offer to take action a_i. Each player secretly and independently chooses at a moment's time t action option a_i, possibly different from the proposed variant, and receives a current payoff $r^i(s_t,a_t)$, which is a function of the current state of the system s_t and the combined option $a_t \in A$. The environment then moves to a new state s_{t+1} according to the probability distribution $p(s_{t+1} \mid s_t,a_t)$ and the process is repeated at time $t+1$.

Correlated equilibrium is determined by the united distribution of player strategies $\sigma \in \Delta(A)$, when each agent does not have the motivation to deviate unilaterally [33]:

$$\sum_{a_{-i}\in A_{-i}} \sigma^{CE}(a_{-i} \mid a_i)Q^i[s',(a_{-i},a_i)] \geq \sum_{a_{-i}\in A_{-i}} \sigma^{CE}(a_{-i} \mid a_i)Q^i[s',(a_{-i},\widetilde{a}_i)],$$

where $A_{-i} = \underset{j=1,\, j\neq i}{\overset{L}{\times}} A_j$, $A = A_{-i} \times A_i$, $a_{-i} \in A_{-i}$, $a = (a_{-i},a_i) \in A$, $a_i,\widetilde{a}_i \in A_i$, $\sigma^{CE}(a_i) = \sum_{a_{-i}\in A_{-i}} \sigma^{CE}(a_{-i},a_i)$, $\sigma^{CE}(a_{-i} \mid a_i) = \sigma^{CE}(a_{-i},a_i)\big/\sigma^{CE}(a_i)$, $\sigma^{CE}(a_i) > 0$.

All Nash equilibrium points are correlated equilibrium points. If $\forall i \in I$ $\sigma^{CE}(a_{-i} \mid a_i) = \sigma^{CE}(a_{-i} \mid \widetilde{a}_i)$, $\forall a_i,\widetilde{a}_i \in A_i$, $\forall a_{-i} \in A_{-i}$, $\forall \sigma^{CE}(a_i),\sigma^{CE}(\widetilde{a}_i) > 0$, then correlated equilibrium is also Nash equilibrium. To solve the game by method (8), the cost operator $V_t^i(s_{t+1})$ the state of the system is determined by the point σ^{CE} correlated equilibrium: $CE(Q_t^i(s_{t+1})) = \sum_{a\in A} \sigma^{CE}(a)Q_t^i(s',a)$. The set of CE equilibrium points is non-empty, convex and compact and can be effectively calculated using linear programming methods. In the case of maximizing total player winnings, the task of linear programming is to find σ^{CE} can be formulated as follows:

$$\sum_{a\in A}\sigma(a)\sum_{i=1}^{L} Q^i(s,a) \to \max_{\sigma}, \quad \sum_{a_{-i}\in A_{-i}} \sigma(a_{-i},a_i)\left(Q^i[s',(a_{-i},a_i)] - Q^i[s',(a_{-i},\widetilde{a}_i)]\right) \geq 0,$$

$$\forall i \in I, \forall a_i \in A_i, \forall \tilde{a}_i \in A_i, \forall s \in S,\ \sigma(a) > 0\ \ \forall a \in A,\ \sum_{a \in A} \sigma(a) = 1.$$

Based on the distribution $\sigma(a)\ \ \forall a \in A$ players' own strategies are determined $\pi_i\ \forall i \in I$. There are various options for switching from σ to π_i depending on the type of strategies and players' level of awareness. For example, pure strategies determine the maximum value of the operator $CE(Q^i(s))$:

$$a_i = \arg\max_{\sigma} CE(Q^i(s)).$$

Taking into account that $\sum_{a_i \in A_i} \sigma(a_i) = 1$, it can be assumed that mixed strategies $\pi_i(a_i) = \sigma(a_i) = \sum_{a_{-i} \in A_{-i}} \sigma(a_{-i}, a_i)$, or:

$$\pi_i(a_i) = \sum_{a_{-i} \in A_{-i}} \sigma(a_{-i}) Q^i(s, a_{-i}, a_i) \Big/ CE(Q^i(s)).$$

Pareto Equilibrium (PE) optimality occurs in the Common-Interest Markov Game, where the payoff matrices are the same for all players $Q_t^i(s,a) = Q_t^j(s,a)\ \forall i, j \in I$, $\forall s \in S,\ \forall a \in A$ [33]. A game with different payoff matrices can be turned into a game of shared interests by a convolution

$$PE\left(Q_t\left(s_{t+1}\right)\right) = \sum_{k=1}^{L} \lambda_k \sum_{a \in A} Q_t^k(s', a) \prod_{j=1}^{L} \pi_j^{PE}(s', a_j),$$

where $\lambda_j > 0\ (j = 1..L)$.

The search for the game's PE solution is done independently by agent strategies, similar to the search for a NE solution. A multi-agent game is optimal for Pareto if there is no common player strategy that improves the winnings of all players: $Q_t^i(s, \pi^{PE}) \geq Q_t^i(s, \pi)$. Pareto-optimal mixed strategies $\pi^{PE}(s) = \left(\pi_1^{PE}(s), ..., \pi_L^{PE}(s)\right)$ can be obtained by maximizing the convolution of concave (up) payoff functions:

$$\sum_{k=1}^{L} \lambda_k \sum_{a \in A} Q_t^k(s', a) \prod_{j=1}^{L} \pi_j(s', a_j) \to \max_{\pi}.$$

To calculate optimal collective strategies $\pi^*(s) = \left(\pi_1^*(s), ..., \pi_L^*(s)\right)$ (NE, CE, PE) agent with number i need to know Q-functions of all agents: $Q(s) = \left(Q_t^1(s), ..., Q_t^L(s)\right)$. In the absence of such information, each agent should evaluate the value Q - functions in the learning process. For this i -agent monitors the current gains of other agents and modifies their estimates Q-functions according to (8). To ensure that method (8) converges to one of the points of collective equilib-

rium, it is necessary to impose a limit on the rate of change of its adjustable parameters. The general limitations are as follows [11, 14, 34]:

$$\sum_{t=0}^{\infty}\alpha_t=\infty,\qquad \sum_{t=0}^{\infty}\alpha_t^2<\infty\,, \tag{10}$$

where $\alpha_t=t^{-\kappa}\ (\kappa>0)$ is monotonically decreasing positive sequences of real values.

4 Stochastic Game Solving Algorithm

Step 1. Set the start time $t=0$; the initial values of the payoff matrices $Q_t^i(s,a_1,...,a_L)=\varepsilon\ \ \forall s\in S,\ \forall a_i\in A_i,\ \forall i\in I$, where $0<\varepsilon<<1$ is small positive value; the value of the gain discount parameter $\gamma\in(0,1]$; the initial state of the system s_0.

Step 2. Perform a random selection of agent actions $a=(a_1,...,a_L)$ based on strategies $\pi=(\pi_1,...,\pi_L)$. The value of the strategies can be calculated from the current estimates of the payoff matrices $\forall i\in I$:

$$\pi_i(s,a_i)=\sum_{a_{-i}\in A_{-i}}Q_t^i(s,a_{-i},a_i)\Big/\sum_{a\in A}Q_t^i(s,a)\,,\ \forall a_i\in A_i\,.$$

Step 3. Get current agent payouts $r_t=(r_t^1,...,r_t^L)$.

Step 4. Determine the new state of the system $s_{t+1}=s_t(a_1,...,a_L)$.

Step 5. Calculate function $V_t^i(s_{t+1})$ according to the specific condition of collective equilibrium (NE, CE, PE).

Step 6. Modify the payoff matrix $Q_{t+1}=\left(Q_{t+1}^i(s_t,a_t)\,|\,i=1..L\right)$ according to (8).

Step 7. If $\left\|Q_{t+1}^i-Q_t^i\right\|<\varepsilon\ \ \forall i=1..L$, then ask $t:=t+1$ and go to step 2.

Step 8. Print the calculated values of the payoff matrices $Q(s)=\left(Q^1(s),...,Q^L(s)\right)$ and strategies $\pi(s)=(\pi_1(s),...,\pi_L(s))\ \ \forall s\in S$. End of algorithm.

5 The Results of Computer Simulation

Let's solve the stochastic game of two agents with two pure strategies in a two-state environment. The matrices of the average payoffs of such a game are given in Table 1.

Table 1. Player gains matrix

States	Strategies	The first player		The second player	
	×	$\pi_2(s_1, a_2[1])$	$\pi_2(s_1, a_2[2])$	$\pi_2(s_1, a_2[1])$	$\pi_2(s_1, a_2[2])$
s_1	$\pi_1(s_1, a_1[1])$	0.4	0.1	0.9	0.2
	$\pi_1(s_1, a_1[2])$	0.1	0.9	0.2	0.9
	×	$\pi_2(s_2, a_2[1])$	$\pi_2(s_2, a_2[2])$	$\pi_2(s_2, a_2[1])$	$\pi_2(s_2, a_2[2])$
s_2	$\pi_1(s_2, a_1[1])$	0.4	0.6	0.5	0.2
	$\pi_1(s_2, a_1[2])$	0.6	0.8	0.6	0.7

Under uncertainty, the elements of the average payoff matrix $\left[v^i(s,a)\right]_{\substack{\forall s\in S\\ \forall a\in A}}$ a priori unknown and available for observation in the form of random current values

$$r^i(s,a) = Normal(v^i(s,a), d^i(s,a)),$$

distributed by normal law with mathematical expectation $v^i(s,a)$ and dispersion $d^i(s,a)$. Normally distributed random variables are obtained by summing twelve evenly distributed random numbers $\omega \in [0, 1]$:

$$r^i(s,a) = v^i(s,a) + \sqrt{d^i(s,a)}\left(\sum_{j=1}^{12}\omega_j - 6\right), \tag{11}$$

where $d^i(s,a) = d > 0 \ \forall s \in S, \forall a \in A$. If at time t the system was in a state of disre-pair $s \in S$, then after implementing pure strategies $a = (a_1,...,a_L)$, where $a \in A = \underset{i=1}{\overset{L}{\times}} A^i$, agents receive current winnings $r_t^i(s,a)$, calculated according to (11).

After receiving current winnings, each agent lists the corresponding item Q-matrix according to the algorithm modified for uncertainty conditions BR:

$$Q_{t+1}^i(s,a_1,...,a_L) = (1-\alpha_t)Q_t^i(s,a_1,...,a_L) + \alpha_t(r_t^i + \gamma \max_{a_i} Q_t^i(s,a_1,...,a_L)). \tag{12}$$

Based on Q-matrices the current values of mixed strategies are calculated using the Boltzmann method:

$$\pi_i(a_i(k)\,|\,s) = e^{Q_i^*(s,a_i(k))/T} \Big/ \sum_{j=1}^{N_i} e^{Q_i^*(s,a_i(j))/T}, \ k = 1..N_i, \tag{13}$$

where $Q_i^*(s,a_i(k)) = \max\limits_{a_{-i}} r^i(a_{-i},a_i(k))$, $a_{-i} \in A_{-i}$, $A_{-i} = \underset{\substack{j=1 \\ j\neq i}}{\overset{L}{\times}} A^j$, $T > 0$ is temperature coefficient.

Vector elements of mixed strategies π_i define a discrete distribution by which the values of random pure strategies are determined i-agent at the next time:

$$a_i(s) = \left\{ A^i(s,k) \middle| k = \arg\left(\min_k \sum_{j=1}^{k} \pi_i(s,a_i(j)) > \omega \right), k = 1..N_i \right\}, \ \forall s \in S, \ \forall i \in I, \quad (14)$$

where $\omega \in [0, 1]$ is random variable with uniform distribution.

The change of states of the dynamic system is determined by the discrete distribution $p(s' \mid s,a) = p \ \ \forall s \in S, \forall a \in A$:

$$s = \left\{ S(k) \middle| k = \arg\left(\min_k \sum_{j=1}^{k} p(j) > \omega \right), k = 1..M \right\}. \quad (15)$$

Let the states change $s \in S$ system is implemented with equal probabilities $p(s' \mid s,a) = (|S|_k^{-1}, k = 1..M) \ \ \forall s \in S, \forall a \in A$, that is $p(s' \mid s,a) = (0.5; 0.5)$ for $|S| = 2$. Agents' average payoffs are calculated taking into account the transition probabilities of the medium from one state to another: $V^i = \sum\limits_{s \in S} p(s) V^i(s)$, $i = 1..L$, where $V^i(s) = \sum\limits_{a \in A} v^i(s,a) \prod\limits_{j=1}^{L} \pi_j(s,a)$ is average agent gain in the state $s \in S$.

The trajectories of changing agent strategies within a unit simplex and the appearance of average payoff functions $V^i(s)$, which correspond to the data of the table 1, is shown in Fig. 1 and Fig. 2.

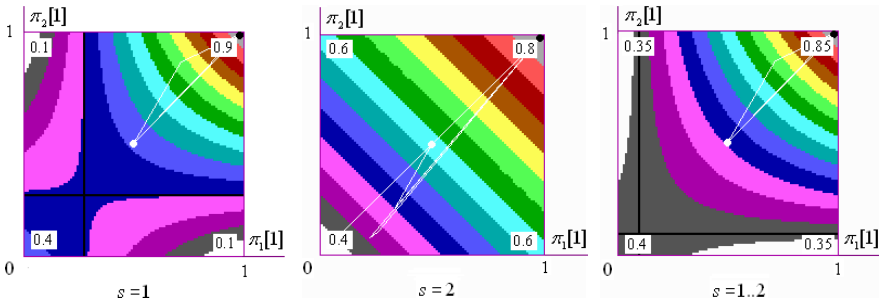

Fig. 1. First agent average payoff functions

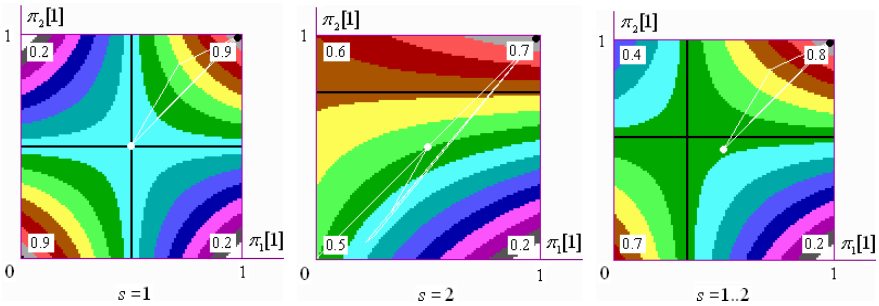

Fig. 2. Second agent average payoff functions

The BR method (12 – 15) ensures that stochastic play is solved at the vertices of a unit simplex with the maximum value of the mean gain function. The percentage of options for achieving optimal game resolution depends on the absolute difference between the two largest consecutive values of the average payoff features.

The convergence of the method is estimated by the error of fulfillment of the complementary slackness condition [35], weighted by mixed strategies: $\Delta = L^{-1}\sum_{i\in I}\left\|\pi_i - \tilde{\pi}_i\right\|^2$, where $\tilde{\pi}_i = diag(\pi_i)\nabla \mathrm{V}^i / V^i$; $diag(\pi_i)$ is diagonal square matrix of order N_i, formed from vector elements π_i; $\nabla V^i = \left(V^i[j] \middle| j = 1..N_i\right)$ is vector median payoff function for fixed net strategies i-player; $V^i = \sum_{j=1}^{N_i} V^i[j]\pi_i[j]$ is average payoff function i-player; $\|*\|$ is Euclidean vector norm. The complementary slackness condition characterizes the gameplay in Nash mixed strategies. The weighted condition additionally takes into account the game's solutions in pure strategies. Average payoff function graphs Υ and norms for deviating mixed strategies from their target values Δ filed in Fig. 3.

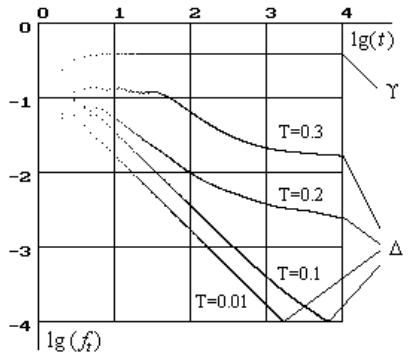

Fig. 3. Characteristics of the convergence of the game Q-method

The gains in Fig. 3 are averaged over the number of players: $\Upsilon = L^{-1}\sum_{i=1}^{L}\Upsilon_i$, where $\Upsilon_i \geq 0$ is discounted current gains i -player. Deviation schedule drop Δ mixed strategies from their target indicates the convergence of the game Q -method.

The value of the temperature coefficient T has a significant impact on the convergence of the game method. The rate of convergence is determined by the rapid decline of the function graph Δ, which can be estimated by the value of the acute angle of linear approximation of the function graph Δ with the time axis. With the growth T rate of convergence of game Q -method decreases.

6 Conclusions

The promotional training method (8) considered in the deterministic version requires the knowledge of each agent Q -functions of all other agents. These functions are used by agents to identify strategies that provide method dynamics toward the points of collective equilibrium. Value Q -functions can be obtained by exchanging information between agents. If integrated information about Q -functions are not available to the agent, then he must determine their value independently in the learning process, observing the current benefits of other agents and performing evaluations Q -functions according to (8). If such observations are not possible, the agents may perform reflective assessments Q -functions of other agents.

Another method of constructing incentive training algorithms for agents under uncertainty is to apply the stochastic approximation method to the corresponding collective equilibrium condition.

The practical use of game-based promotional training methods requires their prior analysis to determine the conditions for convergence to a state of collective equilibrium. Such studies are based on the evaluation of sequences of random variables, which characterize the current deviations of players' strategies from their optimal values.

The rate of convergence of the game method of Q-learning is determined by the parameters α_t and T. Parameter α_t must satisfy the general conditions of stochastic approximation (10). The value of the parameter T depends on the absolute values of the elements of Q-matrices. It is experimentally established that for the given matrices of average payoffs the convergence of the game Q-method is provided at $T \in (0, 0.2]$ in the parameter value range $\alpha_t = t^{-\kappa}$, $\kappa \in (0, 1]$. The highest convergence rate of the game method of promotional learning is achieved at $T = 10^{-2}$.

References

1. Weiss, G.: Multiagent Systems. Second Edition. The MIT Press. (2013)

2. Lee, Y., Chong, Q.: Multi-agent Systems Support for Community-Based Learning. Interacting with Computers, 15(1), 33 – 55. (2003)
3. Kravets, P.: The Methodology of Multi-Agent Systems: a Modern State and Future Trends. In: Proceedings of the International Conference on computer science and information technologies – CSIT'2006, 125 – 127. (2006)
4. Dignum, F., Bradshaw, J., Silverman, B.G., Doesburg, W.: Agent for Games and Simulations: Trends in Techniques, Concepts and Design. Springer. (2009)
5. Kravets, P.: The Control Agent with Fuzzy Logic. In: Perspective Technologies and Methods in MEMS Design, MEMSTECH, 40 – 41. (2010)
6. Scerri, P., Vincent, R., Mailler, R.T.: Coordination of Large-Scale Multiagent Systems. Springer. (2010)
7. Byrski, A, Kisiel-Dorohinicki, M.: Evolutionary Multi-Agent Systems: From Inspirations to Applications. Springer. (2017)
8. Radley, N.: Multi-Agent Systems – Modeling, Control, Programming, Simulations and Applications. Scitus Academics LLC. (2017)
9. Yang, S., Xu, J.-X., Li, X., Shen, D. : Iterative Learning Control for Multi-Agent Systems Coordination. Wiley-IEEE Press. (2017)
10. Sun, Z.: Cooperative Coordination and Formation Control for Multi-Agent Systems. Springer. (2018)
11. Nazin, A. V., Poznyak, A. S.: Adaptive Choice of Variants: Recurrence Algorithms (in russian). Moscow: Science. (1986)
12. Kaelbling, L., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. In: Journal of Artificial Intelligence Research, 4, 237 – 285. (1996)
13. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press. (1998)
14. Watkins, C.J.C.H., Dayan, P.: Q-Learning. In: Machine Learning, Kluwer Academic Publishers, Boston, 8, 279 – 292. (1992)
15. Ummels, M.: Stochastic Multiplayer Games: Theory and Algorithms. Amsterdam University Press. (2014)
16. Ungureanu, V.: Pareto-Nash-Stackelberg Game and Control Theory: Intelligent Paradigms and Applications. Springer. (2018)
17. Zheng, J., Cai, Y., Wu, Y., Shen, X.: Dynamic Computation Offloading for Mobile Cloud Computing: A Stochastic Game-Theoretic Approach. In: IEEE Transaction on Mobile Computing, 18(4), 771 – 786. (2018)
18. Chen, B.-S.: Stochastic Game Strategies and their Applications. CRC Press. (2019)
19. Kravets, P., Pasichnyk, V., Kunanets, N., Veretennikova, N. Game Method of Event Synchronization in Multiagent Systems. In: Advances in Intelligent Systems and Computing (AISC), ICCSEEA 2019 – Proceedings , 938, 378 – 387. (2019)
20. Kravets, P., Burov, Y., Lytvyn, V, Vysotska V.: Gaming Method of Ontology Clusterization. In: Webology, 16(1), 55 – 76. (2019)
21. Gao, J., You, F.: A Stochastic Game Theoretic Framework for Decentralized Optimization of Multi-Stakeholder Supply Chain Under Uncertainty. In: Compute & Chemical Engineering, 122, 31 – 46. (2019)
22. Lalropuia, K. C., Gupta, V.: Modeling Cyber-Physical Attacks Based on Stochastic Game and Markov Processes. In: Reliability Engineering & System Safety, 181, 28 – 37. (2019)
23. Lozovanu, D. Pure and Mixed Stationary Nash Equilibria for Average Stochastic Positional Games. In: Frontiers of Dynamic Games, 131 – 155. (2019)
24. Garrec, T. Communicating Zero-Sum Product Stochastic Games. In: Journal of Mathematical Analysis and Applications, 477(1), 60 – 84. (2019)

25. Kloosterman, A. Cooperation in Stochastic Games: a Prisoner's Dilemma Experiment. In: Experimental Economics, 1 – 21. (2019)
26. Lin, X., Adams, S. C., Beling, P. A. Multi-Agent Inverse Reinforcement Learning for Certain General-Sum Stochastic Games. In: Journal of Artificial Intelligence Research, 66, 473 – 502. (2019)
27. Saldi, N., Basar, T., Raginsky, M. Approximate Nash Equilibria in Partially Observed Stochastic Games with Mean-Field Interactions. In: Mathematics of Operations Research, 44(3), 1006 – 1033. (2019)
28. Yamamoto, Y. Stochastic Games with Hidden States. In: Theoretical Economics, 14, 1115 – 1167. (2019)
29. Fudenberg, D., Levine, D.K.: The Theory of Learning in Games. Cambridge. (1998)
30. Puterman, M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, New York. (2005)
31. Hu, J., Wellman, M. P., Nash Q-learning for general-sum stochastic games. In: Machine Learning Research, 4, 1039 – 1069. (2003)
32. Weinberg, M., Rosenschein, J.S.: Best-Response Multiagent Learning in Non-Stationary Environments. In: AAMAS'04, New York, USA. (2004)
33. Greenwald, A., Hall, K.: Correlated Q-learning. In: Proceedings of the Twentieth International Conference on Machine Learning, 242 – 249. (2003)
34. Kushner, H., Yin, G. G.: Stochastic Approximation and Recursive Algorithms and Applications. Springer Science & Business Media. (2013)
35. Neogy, S. K., Bapat, R. B., Dubey, D.: Mathematical Programming and Game Theory. Springer. (2018)