

# Game Task of Ontological Project Coverage

Petro Kravets, Vasyl Lytvyn, Victoria Vysotska, Yevhen Burov and Ivanna Andrusyak

Lviv Polytechnic National University, S. Bandera street, 12, Lviv, 79013, Ukraine

## Abstract

In this work, a multi-agent game model is developed to form virtual teams of project implementers based on subject ontology libraries. The competencies and abilities of agents required to execute projects are defined by subsets of ontologies from ontology library. Intelligent agents randomly, simultaneously and independently select one of the projects at discrete times. Agents that choose the same project determine the current composition of the team for its execution. For agent teams, the current loss is calculated for not covering the required project competencies by the agents' combined capabilities. This penalty is used to adaptively adjust mixed player strategies. The likelihood of formation of teams whose current composition has led to the reduction in the penalty for not covering ontologies is increasing. During the repetitive stochastic game, agents form vectors of mixed strategies that minimize average losses for non-coverage of projects. To solve the problem of game-based project coverage, an adaptive recurrent Markov method is developed based on a stochastic approximation of the modified condition of complementary non-rigidity, which holds in Nash equilibrium points. Computer simulation has confirmed the viability of using stochastic game model to form project teams having the necessary ontological support under uncertainty. The convergence of the game method is ensured by conforming to fundamental conditions and limitations of stochastic optimization. The validity of the experimental results is confirmed by the repeatability of results obtained for different sequences of random variables.

## Keywords 1

Multi-agent system, ontology, project, stochastic game, adaptive game method.

## 1. Introduction

In today's information society with advanced means of communication through mobile devices and computer networks, the formation of various virtual organizations and communities is relevant. Such virtual associations of people for professional or other interests are intended to solve promptly and efficiently a variety of problems, such as completing regular project tasks, creating start-ups for attracting investors, organizing network marketing or distance learning, solving complex issues in science, economics and public administration, building various Internet services, discussing political and social processes, etc. [1 - 3].

Virtual communities as project execution teams provides such advantages as geographical distribution, decentralization, lack of departmental barriers, high mobilization ability, integration of the best experience and modern technologies for the implementation of project, the possibility to attract specialists with diverse expertise, resulting in the creation of favorable conditions for professional partnership. Furthermore, they provide better coordination of efforts to achieve the goal, competitiveness, prompt resolution of urgent problems, flexibility of structure and function, adaptability to changes in the outside world, remote access to computing and information resources.

---

Proceedings of the 2nd International Workshop IT Project Management (ITPM 2021), February 16-18, 2021, Slavsko, Lviv region, Ukraine  
EMAIL: Petro.O.Kravets@lpnu.ua (P. Kravets); vasyi.v.lytvyn@lpnu.ua (V. Lytvyn); victoria.a.vysotska@lpnu.ua (V. Vysotska); yevhen.v.burov@lpnu.ua (Y. Burov); andrusyak.ivanna@gmail.com (I. Andrusyak)  
ORCID: 0000-0001-8569-423X (P. Kravets); 0000-0002-9676-0180 (V. Lytvyn); 0000-0001-6417-3689 (V. Vysotska); 0000-0001-8653-1520 (Y. Burov); 0000-0001-6601-4374 (I. Andrusyak)



© 2021 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).  
CEUR Workshop Proceedings (CEUR-WS.org)

The modeling of the dynamics of virtual associations in a distributed environment can be performed using multi-agent systems [4 - 7]. An agent is an informational object with elements of artificial intelligence that can make autonomous decisions, interact with other agents and a human person to achieve its goal. A group of such agents that solve a common problem in a computer information network is called a multi-agent system.

In order to accomplish their task the agents need to have some knowledge in one or more subject areas represented formally in the form of ontologies. Interacting with each other, agents can query, compare ontologies, and integrate them to gain new knowledge, intersect ontologies to discover shared knowledge, enrich or correct them. [8 - 12].

Generally, agents' knowledge is highly specialized. Several ontologies, which describe different subject areas, are usually required to complete the project. On the other hand, for the full intellectual and informational support of the project, the agents must be able to form teams (communities, groups, coalitions). A team is a community of agents formed to reach a goal or to accomplish a task using shared knowledge and collaboration. In order for the project to be successful, the combined capabilities of the agent team in knowledge of domain-specific ontologies must cover the competencies required to complete the project. In addition, the team facilitates the organization and coordination of agents, decreases the complexity of the communication process and reduces the reaction time when changes in the information environment occur.

In order to create a team, agents must be able to find and identify each other on the network by negotiating goals and attributes. Centralized team formation limits agent autonomy and is problematic when using distributed data sources. Ideally, ontology agents should be able to group themselves based on self-organizing mechanisms because their coordinated interaction and the application of adaptive decision-making rules using only local information [13 - 15].

Ontological project support is a dynamic process with elements of uncertainty, coming from not clearly defined goals, uncertain initial data, changes in the process of project implementation, development of ontologies over time, imperfect knowledge of project contractors, uncontrolled external factors. [16 - 19]. Therefore, ontology agents involved in project implementation must be built as adaptive, self-learning systems.

Multi-agent support for virtual communities has received considerable attention in the current scientific literature [20 - 22]. However, the problem of adaptive ontology project coverage based on the formation of teams of agents with problem-oriented knowledge is not sufficiently researched.

Project coverage belongs to the class of NP-complex combinatorial optimization problems. Approximate algorithms, such as greedy, genetic, ant colony, artificial neural networks, and others [23 - 28] are used to solve such problems in the allowed polynomial time.

In this paper, we propose a new method of approximating the solution of the problem of project coverage based on the results of stochastic game theory [29 - 31]. The formation of agent teams to execute projects is formulated as a competitive task of securing an agent for one of the projects. In the process of finding the optimal coverage, it is possible to move an agent from one project to another, which may temporarily disrupt the ontological support of the project and make it impossible to execute. Competition problems are studied by game theory and under uncertainty by stochastic game theory. A discrete deterministic game can be solved in a finite number of computational steps. Discrete stochastic repetitive game unfolds for an unlimited period. This game provides multi-step adaptive search for one of the solutions of the problem with a given accuracy and within practically acceptable time limit. It can be used to solve deterministic or stochastic multiple-criteria optimization problems, but it is especially useful and efficient in stochastic uncertainty conditions, when a complete iteration over variants cannot be performed due to random response of a controlled system to the choice of the same strategy at different times. The adaptive stochastic game mechanism compensates for the lack of a priori information by collecting and processing current data at each step of the game. Considering the presence of competitive goals and a priori uncertain factors in project management, it is important to apply stochastic game methods for ontological support of projects.

***The purpose of this work*** is to develop a self-learning game model for ontological project support by forming teams of agents in an uncertain environment.

To achieve this goal it is necessary to do the following:

- Formulate a stochastic game problem of covering projects by ontology agents,
- Create an adaptive method and algorithm for game problem solution,

- Develop a computer program model of game-based selection of ontology agents to execute projects,
- Analyze and discuss the results obtained.

## 2. Mathematical model of stochastic game

Let's start with a library of ontologies  $\Omega = \{O_1, O_2, \dots, O_q\}$ , where each element specifies knowledge in specific subject area. It is required to organize the development of  $m$  projects  $\Pi = \{\Pi_1, \Pi_2, \dots, \Pi_m\}$  with corresponding ontological support. Each project has an associated set of ontological knowledge (competence)  $\Pi_i = \{O_1, O_2, \dots, O_r\} \subseteq \Omega$ , required for its execution.

The set of agents  $A = \{A_1, A_2, \dots, A_n\}$ ,  $n \geq m$  defines the qualified labor on the market. The capabilities of each agent are determined by a set of ontologies  $A_i = \{O_1^i, O_2^i, \dots, O_s^i\} \subseteq \Omega$ . In general case,  $A_i \cap A_j \neq \emptyset$ , so the agents can have the same capabilities in one or several knowledge areas.

Let's assume the completeness of the capabilities for the set of agents and set of competencies required to execute projects. Without loss of generality, we will assume also that the knowledge of all agents covers the library of ontologies  $\bigcup_{A_i \in A} A_i = \Omega$ , required for projects execution  $\bigcup_{\forall \Pi_k \in \Pi} \Pi_k = \Omega$ .

Therefore, the agents' ontological knowledge is sufficient for all projects execution.

We will need to form a set of virtual agent teams  $\Pi' = \{G_1, G_2, \dots, G_m\}$  for the execution of all projects. Each team corresponds to the agent group  $G_k = \{A_1^k, A_2^k, \dots, A_g^k\}$ ,  $k = 1..m$ , where  $\bigcup_{k=1..m} G_k = A$ ,  $G_k \cap G_j = \emptyset$ . The capabilities of the agent teams must meet the competency requirements for the respective projects.

The formation of virtual teams of agents will be performed by the stochastic game method, which is specified by a tuple

$$(A, U^i, \Xi^i \mid \forall A_i \in A), \quad (1)$$

where  $A$  – is the set of agents or players;  $U^i = \{u_1^i, u_2^i, \dots, u_m^i\}$  – is the set of pure strategies of player  $i$ , which determines his affiliation with one of the teams;  $\Xi^i : U \rightarrow R^1$  – the cost function of player  $i$ ;  $U = \times_{A_i \in A} U^i$  – set of combined strategies obtained by the joint choice of all players..

Agents can choose one of the teams themselves. Possible choices are given by strategy vectors  $U^i$ . The choices are made independently and randomly at times  $t = 1, 2, \dots$ . An agent  $A_i \in A$  is included in group  $G_k$ , if its selected pure strategy  $u_t^i$  corresponds to the strategy of the group  $u_t^k$ :

$$G_k = \bigcup_{\forall A_i \in A} \chi(u_t^i = u_t^k) \cdot A_i, \quad k = 1..m, \quad (2)$$

where  $\chi(*) \in \{0, 1\}$  – indicator event function,  $1 \cdot A_i = A_i$ ,  $0 \cdot A_i = \emptyset$ .

A prerequisite for successful completion of the project is its full ontological support by a team of agents. The capabilities of the agent team should cover the competencies required to complete the project:

$$\bigcup_{\forall A_j^k \in G_k} A_j^k \supseteq \Pi_k, \quad k = 1..m. \quad (3)$$

It is advisable to ensure the perfect coverage of all projects, when  $\Pi' = \Pi$ .

For the breach of project coverage, a penalty is assigned, which the relative number of uncovered project ontologies (dimensionless value) measures:

$$\xi_t^k [1] = \left| \Pi_k \setminus \bigcup_{\forall A_j^k \in G_k} A_j^k \right| / |\Pi_k|, \quad k = 1..m, \quad (4)$$

where  $|*|$  – is the cardinality of the set,  $\Pi_i \neq \emptyset$ ,  $\xi_t^k [1] \in [0, 1]$ .

The formula (4) does not rule out the possibility that all agents will select one or more projects and the remaining projects will remain uncovered. Thus, by selecting only one of the projects, the agents knowingly (due to the completeness of the ontology library's coverage of the agents' capabilities and competencies required to execute the projects) ensure its coverage and receive a minimum loss, which further encourages them to remain on the project. Therefore, such an assessment is insufficient when all projects coverage is required.

Since several agents may have knowledge of the same subject areas, the problem of combinatorial optimization appears related to the minimum required project coverage. This problem belongs to the class of NP-complex problems.

Instead of solving the complex problem of minimum coverage of sets, we introduce a payment for deviation from the planned cost of the project (dimensionless relative value):

$$\xi_t^k[2] = (C'(\Pi_k) - C(\Pi_k)) / \max\{C'(\Pi_k), C(\Pi_k)\}, \quad (5)$$

where  $C'(\Pi_k) = \sum_{A_j^k \in G_k} C(A_j^k)$  – is the cost of coverage for project  $k$ ,  $C(A_j^k) > 0$  – is the cost of services of agent  $j$ , taking part in execution of project  $k$  (self-assessment of the agent's abilities in monetary terms),  $C(\Pi_k) > 0$  – is the cost of execution of project  $k$ ,  $\xi_t^k[2] \in [-1, 1]$ .

If  $\xi_t^k[2] < 0$ , then we have an incentive (a negative penalty), otherwise – a penalty. This encourages the selection of a team of agents with a minimum total cost of services offered.

If the total cost of the offered services exceeds the planned cost of the project, the selection of such team members will be penalized. Thus we avoid the ontological over-coverage of the project by team of the contractors (within the project's estimated cost), or otherwise by duplication of the capabilities of the agents who selected the specific project.

The combined fine for failure to organize the project  $\Pi_k$  implementation consists of penalties (4) and (5):

$$\zeta_t^k = \lambda \xi_t^k[1] + (1 - \lambda) \xi_t^k[2] + \mu_t, \quad (6)$$

where  $\lambda \in [0, 1]$  – is the weight coefficient,  $\mu_t$  – is the random value (additive white Gaussian noise), which reflects the stochastic uncertainty of the task. If  $\mu_t = 0 \quad \forall t = 1, 2, \dots$ , then current loss  $\zeta_t^k \in [-1, 1]$ .

All agent of the team  $G_k$ , taking part in the execution of project  $\Pi_k$ , obtain the same current loss (6):

$$\zeta_t^i = \zeta_t^k \quad \forall A_i \in G_k, \quad \forall G_k \in \Pi'. \quad (7)$$

We assume that random losses  $\{\zeta_t^i(u)\}$  of players are independent  $\forall u \in U$ ,  $i = 1..m$ ,  $t = 1, 2, \dots$ , have a constant mean value  $E\{\zeta_t^i(u)\} = v(u) = const$  and limited second moment  $\sup_t E\{[\zeta_t^i(u)]^2\} = \sigma^2(u) < \infty$ . The stochastic parameters of random losses are not known by players a priori.

The course of the game is evaluated by the functions of the average losses of the agents:

$$Z_t^i = \frac{1}{t} \sum_{\tau=1}^t \zeta_\tau^i = \lambda \Xi_t^i[1] + (1 - \lambda) \Xi_t^i[2] + M_t \quad \forall A_i \in A, \quad (8)$$

where  $\Xi_t^i[1] = \frac{1}{t} \sum_{\tau=1}^t \xi_\tau^i[1]$  – is the average losses function for insufficient ontological support for the project,  $\Xi_t^i[2] = \frac{1}{t} \sum_{\tau=1}^t \xi_\tau^i[2]$  – is the average losses function for deviation from the projected cost of

the project,  $M_t = t^{-1} \sum_{\tau=1}^t \mu_\tau$  – is current average value for random noise or current estimate of the random noise mean value (for Gaussian white noise  $\lim_{t \rightarrow \infty} M_t = 0$ ).

The goal of each player is to minimize their own function of the average losses (8) over time:

$$\lim_{t \rightarrow \infty} \overline{Z}_t^i \rightarrow \min \quad \forall A_i \in A. \quad (9)$$

Thus, the stochastic game of ontological support for projects is as follows. By calculating current losses  $\{\zeta_t^i\}$ , each player  $A_i \in A$  should learn how to choose the pure strategies  $\{u_t^i\}$  to ensure that the criteria system (9) is met over time:  $t = 1, 2, \dots$ .

The quality of the formation of agent teams in game is evaluated by the following characteristics:

1) the system function of the average losses of a multi-agent system:

$$Z_t = \frac{1}{n} \sum_{i=1}^n Z_t^i = \lambda \Xi_t[1] + (1 - \lambda) \Xi_t[2] + M_t, \quad (10)$$

where  $n$  – is the number of agents,  $\Xi_t[1] = \frac{1}{n} \sum_{i=1}^n \Xi_t^i[1]$  – is the systemic component of losses

reflecting the lack of project coverage,  $\Xi_t[2] = \frac{1}{n} \sum_{i=1}^n \Xi_t^i[2]$  – is the systemic component of losses reflecting the deviation from project cost balance;

2) the average norm for players' mixed strategies:

$$\Delta_t = \frac{1}{nt} \sum_{\tau=1}^t \sum_{i=1}^n \|p_\tau^i\|, \quad (11)$$

where  $\|\cdot\| \in R^1$  – is the Euclidean vector norm.

Game solutions should satisfy one of the conditions of equilibrium, such as Nash, Pareto, or another, depending on the method of strategy sequencing  $\{u_t^i\} \forall A_i \in A$ .

### 3. Stochastic game solving method

We are creating the sequences of pure strategies  $\{u_t^i\}$  necessary for the solution of the game task based on random distributions, obtained from dynamic vectors of mixed strategies  $p_t^i = (p_t^i[1], p_t^i[2], \dots, p_t^i[m]) \quad \forall i \in D$ , which contain the conditional probabilities of agent  $i$  being in team  $k$ :

$$p_t^i[k] = P\{u_t^i = u^i[k] | u_\tau^i, \zeta_\tau^i (\tau = 1, 2, \dots, t-1)\}, \quad k = 1..m, \quad (12)$$

where  $\{u_\tau^i\}$  – is the history of strategies chosen by player  $i$ ;  $\{\zeta_\tau^i\}$  – is the history of losses associated with those choices.

We construct the stochastic game solving method on the basis of a stochastic approximation of the complementary non-rigidity of the deterministic game, which holds for mixed strategies at Nash equilibrium points [32].

To do this, we define the polylinear function of the average losses for the deterministic game:

$$V^i(p) = \sum_{u \in U} v^i(u) \prod_{A_j \in A; u^j \in u} p^j(u^j), \quad (13)$$

where  $v(u) = M\{\zeta_t^i(u)\}$ .

Then the condition of complementary non-rigidity in the vector form will look like:

$$\nabla_{p^i} V^i(p) - e^m V^i(p) = 0 \quad \forall A_i \in A, \quad (14)$$

where  $\nabla_{p^i} V^i(p)$  – is the gradient of polylinear function of average losses;  $e^m = ((1)_k | k = 1..m)$  – is the all-ones vector;  $p \in S^M$  – is a combined mixed strategy set on a unitary simplex  $S^M$  ( $M = m^n$ ).

To account for solutions on the boundaries of unitary simplex, let us weigh the complementary non-rigidity vector by the elements of the mixed strategy vector:

$$\text{diag}(p^i) \left( \nabla_{p^i} V^i(p) - e^m V^i(p) \right) = 0, \quad (15)$$

where  $\text{diag}(p^i)$  – is square diagonal matrix of order  $m$ , comprised of elements of vector  $p^i$   $\forall A_i \in A$ .

Taking in consideration that

$$\text{diag}(p^i) [\nabla_{p^i} V^i - e^m V^i] = E \{ \zeta_t^i [e(u_t^i) - p_t^i] | p_t^i = p^i \}, \quad (16)$$

and using the stochastic approximation method [34 – 36] we get such recurrent expression:

$$p_{t+1}^i = \pi_{\varepsilon_{t+1}}^m \left\{ p_t^i - \gamma_t \zeta_t^i (e(u_t^i) - p_t^i) \right\} \quad \forall A_i \in A, \quad (17)$$

where  $E$  – is the sign for mathematical mean;  $\pi_{\varepsilon_{t+1}}^m$  – is projection on  $m$ -dimensional unitary simplex  $S^m$  [33];  $\gamma_t > 0$  and  $\varepsilon_t > 0$  – is monotonically declining sequences of positive numbers;  $e(u_t^i)$  – is a vector indicating the agent's choice of pure strategy  $u_t^i = u^i \in U^i$ .

The projection  $\pi_{\varepsilon_{t+1}}^m$  on expendable  $\varepsilon_t$ -simplex  $S_{\varepsilon_{t+1}}^m \subseteq S^m$  ensures the adherence to condition  $p_t^i[k] \geq \varepsilon_t, k = 1..m$ , necessary for completeness of statistical information on the choice of pure strategies, and parameter  $\varepsilon_t \rightarrow 0$  is used as an additional control element for the recurrent method convergence.

Parameters  $\gamma_t$  and  $\varepsilon_t$  can calculated as follows:

$$\gamma_t = \gamma t^{-\alpha}, \quad \varepsilon_t = \varepsilon t^{-\beta}, \quad (18)$$

where  $\gamma > 0; \alpha > 0; \varepsilon > 0; \beta > 0$ .

The convergence of mixed strategies (17) to optimal values with probability 1 and in RMS sense is determined by the ratios of parameters  $\gamma_t$  and  $\varepsilon_t$ , those that must satisfy the fundamental conditions of stochastic approximation [33 - 35].

The choice of pure strategy  $u_t^i[k] \quad \forall A_i \in A$  is accomplished by players based on dynamic random distribution (17):

$$k = \arg \left( \min_{k=1..m} \sum_{j=1}^k p_t^i(u_t^i[j]) > \omega \right) \in \{1..m\}, \quad (19)$$

where  $\omega \in [0, 1]$  – is the real random number with a uniform distribution.

Stochastic game begins with untrained mixed strategies with element values  $p_0^i[k] = 1/m$ , where  $k = 1..m$ . Later, the dynamics of vectors of mixed strategies are determined by the Markov recurrent method (17) – (19).

Therefore, in moments of time  $t = 1, 2, \dots$  each player using the mixed strategy  $p_t^i$  selects the pure strategy  $u_t^i$  (19) and up to the time moment  $t+1$  gets current losses  $\zeta_t^i$  (7). After this, it calculates the mixed strategy  $p_{t+1}^i$  according to (17) – (18).

Due to the purposeful dynamic restructuring of mixed strategies based on the processing of current losses, method (17) - (19) provides an adaptive choice of pure strategies over time.

## 4. Stochastic game solving algorithm

Step 1. Set the initial parameter values:

$t = 0$  – is starting time moment;  
 $m$  – is the number of projects, or the number of virtual teams, or the number of pure strategies of game agents;  
 $n$  – is the number of agents;  
 $\Omega = \{O_1, O_2, \dots, O_q\}$  – is the library of ontologies;  
 $\Pi_k = \{O_1, O_2, \dots, O_r\} \subseteq \Omega$ ,  $k = 1..m$  – is the sets of ontological domain knowledge or competencies, needed for project execution;  
 $A_i = \{O_1^i, O_2^i, \dots, O_s^i\} \subseteq \Omega$ ,  $i = 1..n$  – is the sets of ontologies, describing the capabilities of agents;  
 $C(\Pi) = (C(\Pi_1), C(\Pi_2), \dots, C(\Pi_m))$  – is the costs of projects;  
 $C(A) = (C(A_1), C(A_2), \dots, C(A_n))$  – is the costs of services, provided by agents;  
 $U^i = \{u^i[1], u^i[2], \dots, u^i[m]\}$ ,  $i = 1..n$  – is the vectors of pure strategies for agents;  
 $p_0^i = ((1/m)_k | k = 1..m)$ ,  $i = 1..n$  – is the initial values for agent's mixed strategies;  
 $\gamma > 0$  – is parameter of learning step;  
 $\alpha \in (0, 1]$  – is learning step order coefficient;  
 $\varepsilon$  – is parameter of  $\varepsilon$ -simplex;  
 $\beta > 0$  – is expansion order factor for  $\varepsilon$ -simplex;  
 $\lambda \in [0, 1]$  – is weight coefficient;  
 $t_{\max}$  – is the maximal number of steps for method.

Step 2. Choose the pure strategies (teams)  $u_t^i \in U^i$  of agents  $i = 1..n$  according to (19).

Step 3. Calculate the current losses values  $\zeta_t^i$ ,  $i = 1..n$  according to (7).

Step 4. Calculate the values of parameters  $\gamma_t$  and  $\varepsilon_t$  according to (18).

Step 5. Calculate the elements of mixed strategies vectors  $p_t^i$ ,  $i = 1..n$  according to (17).

Step 6. Calculate the quality characteristics  $Z_t$  (10) and  $\Delta_t$  (11) of project coverage.

Step 7. Move to the next time moment  $t := t + 1$ .

Step 8. If  $t < t_{\max}$ , than continue from step 2, otherwise – game is completed.

## 5. A test case

Agents need to be selected to carry out two projects  $\Pi = \{\Pi_1, \Pi_2\}$  with ontological library  $\Omega = \{O_1, O_2, O_3, O_4, O_5\}$ . Knowledge (competencies) required for the execution of projects are given by sets of ontologies:  $\Pi_1 = \{O_1, O_3, O_5\}$ ,  $\Pi_2 = \{O_2, O_4\}$ . The applicants for participation in projects are described by the set of agents  $A = \{A_1, A_2, A_3, A_4, A_5, A_6\}$ . For each agent the subset of ontologies, describing its capabilities is given:  $A_1 = \{O_1, O_2\}$ ,  $A_2 = \{O_2, O_3\}$ ,  $A_3 = \{O_3, O_4\}$ ,  $A_4 = \{O_4, O_5\}$ ,  $A_5 = \{O_1, O_4\}$ ,  $A_6 = \{O_2, O_5\}$ . The planned costs for the implementation of projects are  $C(\Pi) = (10000, 6000)$  money units. The cost of skilled labor in the labor market is determined by this array  $C(A) = (4000, 2500, 1500, 3500, 2500, 2000)$ .

It is needed to form two virtual teams of agents  $\Pi' = \{G_1, G_2\}$ , with ontological knowledge covering every project with project cost constraints  $C'(\Pi) \leq C(\Pi)$ , where  $C'(\Pi)$  – is the array of project coverage costs.

The ontological support of projects problem can have several solutions. For a considered test case such options of project coverage are possible  $\Pi' \supseteq \Pi$ :

- 1)  $G_1 = \{A_1, A_2, A_4\}$ ,  $G_2 = \{A_3, A_5, A_6\}$ ,  $C'(\Pi) = (10000, 6000)$ ;
- 2)  $G_1 = \{A_1, A_3, A_6\}$ ,  $G_2 = \{A_2, A_4, A_5\}$ ,  $C'(\Pi) = (7500, 8500)$ ;
- 3)  $G_1 = \{A_2, A_4, A_5\}$ ,  $G_2 = \{A_1, A_3, A_6\}$ ,  $C'(\Pi) = (8500, 7500)$ ;
- 4)  $G_1 = \{A_3, A_5, A_6\}$ ,  $G_2 = \{A_1, A_2, A_5\}$ ,  $C'(\Pi) = (6000, 9000)$ ;
- 5)  $G_1 = \{A_2, A_4, A_5, A_6\}$ ,  $G_2 = \{A_1, A_3\}$ ,  $C'(\Pi) = (10500, 5500)$ ;
- 6)  $G_1 = \{A_2, A_3, A_5, A_6\}$ ,  $G_2 = \{A_1, A_4\}$ ,  $C'(\Pi) = (8500, 7500)$ ;
- 7)  $G_1 = \{A_2, A_3, A_4, A_6\}$ ,  $G_2 = \{A_1, A_5\}$ ,  $C'(\Pi) = (9500, 6500)$ ;
- 8)  $G_1 = \{A_1, A_4, A_5, A_6\}$ ,  $G_2 = \{A_2, A_3\}$ ,  $C'(\Pi) = (12000, 4000)$ ;
- 9)  $G_1 = \{A_1, A_3, A_5, A_6\}$ ,  $G_2 = \{A_2, A_4\}$ ,  $C'(\Pi) = (10000, 6000)$ ;
- 10)  $G_1 = \{A_1, A_3, A_4, A_6\}$ ,  $G_2 = \{A_2, A_5\}$ ,  $C'(\Pi) = (11000, 5000)$ ;
- 11)  $G_1 = \{A_1, A_2, A_4, A_5\}$ ,  $G_2 = \{A_3, A_6\}$ ,  $C'(\Pi) = (12500, 3500)$ ;
- 12)  $G_1 = \{A_1, A_2, A_3, A_5\}$ ,  $G_2 = \{A_4, A_6\}$ ,  $C'(\Pi) = (10500, 5500)$ ;
- 13)  $G_1 = \{A_1, A_2, A_3, A_4\}$ ,  $G_2 = \{A_5, A_6\}$ ,  $C'(\Pi) = (11500, 4500)$ .

All options provide redundant coverage of both projects by teams of agents representing ontologies. Redundant coverage is determined by external circumstances, because agents have knowledge of more than one subject area.

Options 1 and 9 determine the quasi-optimal solution of the problem because they are constrained by the cost of projects, but in both cases, there is redundancy of project coverage by agent groups:

- option 1:
 
$$\Pi'_1(G_1) = A_1 \cup A_2 \cup A_4 = \{O_1, O_2, O_3, O_4, O_5\} \supset \Pi_1,$$

$$\Pi'_2(G_2) = A_3 \cup A_5 \cup A_6 = \{O_1, O_2, O_3, O_4, O_5\} \supset \Pi_2,$$
- option 9:
 
$$\Pi'_1(G_1) = A_1 \cup A_3 \cup A_5 \cup A_6 = \{O_1, O_2, O_3, O_4, O_5\} \supset \Pi_1,$$

$$\Pi'_2(G_2) = A_2 \cup A_4 = \{O_2, O_3, O_4, O_5\} \supset \Pi_2.$$

In addition, if multiple agents have knowledge of the same subject area, competition may arise within the team for applying this knowledge to the project. Thus, in option 1, agents  $A_1$  and  $A_2$  from group  $G_1$  compete on the ontology  $O_2$  application to execute the project  $\Pi_1$ , and agents  $A_3$  and  $A_5$  from group  $G_2$  compete on the ontology  $O_4$  for project  $\Pi_2$  execution. In option 9, in group  $G_1$  executing project  $\Pi_1$  there is competition for the use of ontology  $O_4$  by agents  $A_1$  and  $A_5$ , the use of ontology  $O_2$  by agents  $A_1$  and  $A_6$ , the use of ontology  $O_4$  by agents  $A_3$  and  $A_5$ . An alternative to competition is the cooperation and mutual enhancement of agents' knowledge in the same subject area. Options 2 – 4, 6, 7 exceed the cost of the project  $G_2$ . In options 5, 8, 10 – 13, the project  $G_1$  implementation cost is exceeded.

The game algorithm must learn how to choose one of the described project coverage options (within their stated cost) by the contracting agents, which have the necessary knowledge the form of ontology sets.

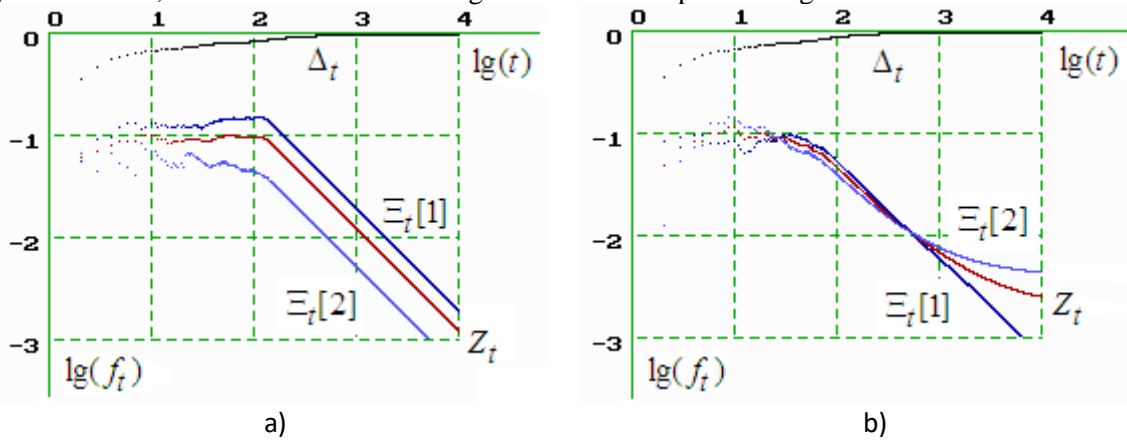
## 6. The results of computer simulation

Computer simulation was performed using game method (17) – (19) with such parameters:  $U^i = \{u_1^i, u_2^i\}$ ,  $\gamma = 1$ ,  $\varepsilon = 0.999/m$ ,  $\alpha = 0.01$ ,  $\beta = 2$ ,  $\lambda = 0.5$ ,  $t_{\max} = 10^4$ .



Fig. 1 and fig. Fig. 2 present, on a logarithmic scale, the graphs of the functions of the average losses of players  $Z_t$ ,  $\Xi_t[1]$ ,  $\Xi_t[2]$ , and the average norm of mixed strategies  $\Delta_t$  that characterize the convergence of the stochastic game of ontological project support. The choice of logarithmic scale is due to the need for a compact representation of simulation results with a large range of values. Exceptions when the current values of the game's characteristic functions are less than zero or equal to zero, were programmatically processed for the logarithmic scale.

As can be seen in Fig. 1 and fig. 2, the game method (17) – (19) minimizes the average losses functions  $Z_t$ ,  $\Xi_t[1]$ ,  $\Xi_t[2]$  over time. The function of average norm of mixed strategies  $\Delta_t$  reaches logarithmic zero, which illustrates how the game is solved in pure strategies



**Figure 1:** The solution of the stochastic game without noise: a) for option 1; b) for option 10

The linear (on logarithmic scale) drop in the graphs of the players' systemic losses functions  $Z_t$ ,  $\Xi_t[1]$ ,  $\Xi_t[2]$  shown on Fig. 1a, indicates the achievement of a quasi-optimal solution of the ontology project coverage problem. The computer simulation shows that in almost 90% of the experiments, methods (17) – (19) provide a quasi-optimal coverage corresponding to options 1 or 9.

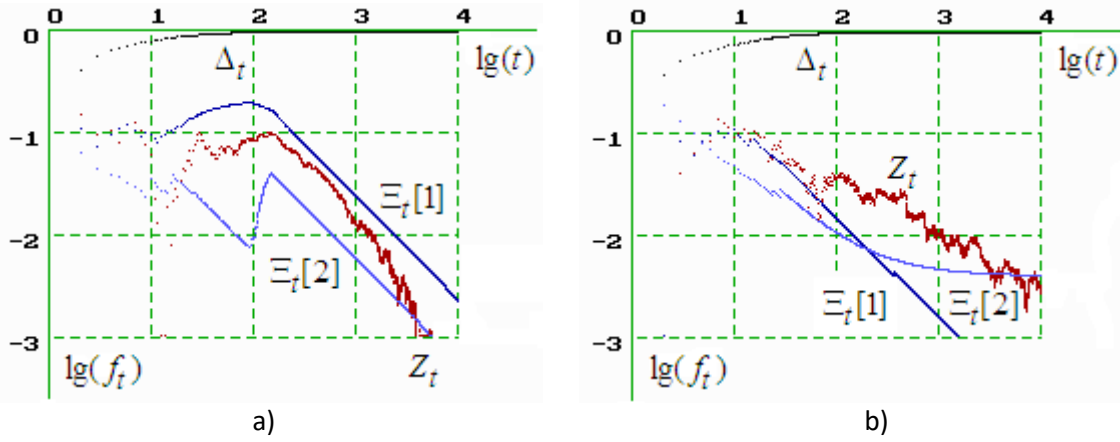
For other permissible coverage options, for example, for option 10, the value of the average loss function  $\Xi_t[1]$  for project coverage deviations decreases linearly, indicating that all projects are covered (Fig. 1b). The value of the average losses function  $\Xi_t[2]$  for disrupting the projects' cost balance goes to a stable value, which indicates the deviation of project implementation costs from the projected cost.

The convergence of game method (17) – (19) to the optimal collective solution depends on the accuracy of its parameters tuning, the ratios of which should satisfy the fundamental requirements of stochastic approximation. It has been experimentally found that reducing the parameter  $\beta \in (0, 2]$  slows down the rate of expansion of the  $\varepsilon$ -simplex and leads to an increase in the number of stochastic game steps required to find one of the project coverage options. A similar effect is observed with the increase of the parameter  $\alpha \in (0, 1]$ , which accelerates the decrease of the search step of the recurrent method. In other words, for the implemented recurrent transformation (17) with the stated method of formation of penalties (7), the expansion of the  $\varepsilon$ -simplex should be fast enough, and the reduction of the search step should be slow. The rapid extension of the  $\varepsilon$ -simplex practically does not limit the magnitude of the search method step. The initially large value of the search step leads to significant dynamics of mixed strategy vectors, enabling players to randomly choose other pure strategies (moving from one project to another), looking for the optimal ontology coverage of the projects. Over time, the magnitude of the search step becomes smaller, and the dynamics of vectors of mixed strategies stabilize, securing the formed agent teams to execute specific projects.

The progress of stochastic game under the conditions of interference is shown in Fig. 2. The stochastic uncertainty of project coverage is given by a normal distribution  $\mu_t \sim Normal(e, d)$  with mathematical expectation  $e = 0$  and variance  $d = 0.25$ . The empirical normal distribution is obtained using the formula:

$$\mu_t = e + \sqrt{d} \left( \sum_{j=1}^{12} \omega_{j,t} - 6 \right), \quad (20)$$

where  $\omega \in [0,1]$  – is a real random number with uniform distribution.



**Figure 2:** The solution of stochastic game in the condition of interference: a) for option 1; b) for option 10

The effect of random noise causes an irregularity in the magnitude of the search step of the recurrent method, which at each step of the stochastic game changes further in proportion to the variance of the interference. This is reflected in the form of a systemic average losses function  $Z_t$ , which behaves as a more pronounced random process.

The additional randomization of current losses with white Gaussian noise with small variance (for example,  $d = 0.25$ ) does not have a significant impact on the learning outcome of the stochastic game. However, increasing the variance of noise causes the solution process to be slowed down or makes it impossible to reach the solution.

## 7. Conclusions

In today's information society with advanced telecommunications through mobile devices and computer networks, it is important to form a variety of virtual organizations and communities. Such virtual associations of people by professional or other interests are designed to quickly solve various tasks: to perform project tasks, create start-ups to attract investors, network marketing, distance learning, solving complex problems in science, economics and public administration, construction of various Internet services, discussion of political and social processes, etc. Objective of the study is to develop an adaptive Markov recurrent method based on the stochastic approximation of the modified condition of complementary non-rigidity, valid at Nash equilibrium points for solving the problem of game coverage of projects. This paper proposes a new self-learning game based method of forming virtual agent teams to execute projects under uncertainty. At the beginning of the game, mixed game agent strategies are untrained and provide an equal choice of projects. The agent training process consists in the purposeful change of vectors of mixed strategies at each step of the game in order to minimize the functions of average losses for insufficient ontological support for projects. At the final stage of learning a stochastic game, stabilization of mixed strategies of agents occurs. The elements of the learned mixed strategies set the probabilities of agents belonging to one of the teams. The result of the game is the formation of teams of agents whose ontological knowledge covers the competencies needed to complete the projects. The adaptive multi-step stochastic game solving method is based on a stochastic approximation of the modified Nash equilibrium complementary non-rigidity condition. The convergence of the game is determined by the fundamental conditions of stochastic approximation and depends on the size of the game (number of players and strategies) and the ratios of the parameters of the recurrent solution method. The proposed method is based on the processing of reactions of the

game environment under a priori uncertainty and therefore has a slow convergence rate, offset by the high computing power of modern computer systems.

In this work the multi-agent game model for formation of virtual teams of executors of projects based on libraries of subject ontologies is developed. The competencies and abilities of agents required to carry out projects are specified by sets of ontologies. Intelligent agents randomly, simultaneously and independently choose one of the projects at discrete times. Agents who have chosen the same project determine the current composition of the team of its executors. For agents' teams, a current penalty is calculated for insufficient coverage of competencies by the combined capabilities of agents. This penalty is used to adaptively recalculate mixed player strategies. The probabilities of selecting those teams whose current composition has led to a reduction in the fine for non-coverage of ontologies are increasing. During the repetitive stochastic game, agents will form vectors of mixed strategies that will minimize average penalties for non-coverage of projects.

For solve the problem of game coverage of projects, an adaptive Markov recurrent method based on the stochastic approximation of the modified condition of complementary non-rigidity, valid at Nash equilibrium points, was developed.

Computer simulation confirmed the possibility of using the stochastic game model to form teams of project executors with the necessary ontological support in conditions of uncertainty. The convergence of the game method is ensured by compliance with the fundamental conditions and limitations of stochastic optimization. The reliability of experimental studies is confirmed by the repeatability of the results obtained for different sequences of random variables.

The disadvantage of the proposed game method is that, in general, it does not guarantee perfect coverage, since it does not provide the elimination of agents whose capabilities are excessive for project execution. Alternatively, as a solution, the additional dummy projects could be created with proposals that will be attractive to the redundant workforce from other projects.

## 8. References

- [1] IGI Global, *Virtual Communities: Concepts, Methodologies, Tools and Applications* (4 Volumes). Information Resources Management Association (USA), 2011. doi: 10.4018/978-1-60960-100-3.
- [2] T. Hutchings, *Real Virtual Community*, volume 35 (2) of *Word & World*, 2015, pp. 151-161.
- [3] A. Roy, *A Typology of Virtual Communities on the Internet: Contingency Marketing Approaches*. Proceedings of the First International Academic Research Conference on Marketing & Tourism, MTCI15Dubai Conference, Dubai-UAE, 22-24 May, 2015, pp. 1-11.
- [4] G. Weiss, *Multiagent Systems*. Second Edition. The MIT Press, 2013.
- [5] A. Byrski, M. Kisiel-Dorohinicki, *Evolutionary Multi-Agent Systems: From Inspirations to Applications*. Springer, 2017. doi: 10.1007/978-3-319-51388-1
- [6] N. Radley, *Multi-Agent Systems – Modeling, Control, Programming, Simulations and Applications*. Scitus Academics LLC, 2017.
- [7] F. Dignum, J. Bradshaw, B.G. Silverman, W. Doesburg, *Agent for Games and Simulations: Trends in Techniques, Concepts and Design*. Springer, 2009. doi : 10.1007/978-3-642-11198-3
- [8] P. Kravets, Y. Burov, V. Lytvyn, V. Vysotska, *Gaming Method of Ontology Clusterization*, volume 16 (1) of *Webology*, 2019, pp. 55-76.
- [9] D. Stuart, *Practical Ontologies for Information Professionals*. Facet Publishing, 2016.
- [10] Y. Aleman, M. J. Somodevilla, *A proposal for domain ontological learning*, volume 133 of *Research in Computing Science*, 2017, pp. 63-70.
- [11] C. M. Keet, *An introduction to Ontology Engineering*, 2018, <http://hdl.handle.net/11427/28312>.
- [12] C. Thomas, *Ontology in Information Science*. IntechOpen, 2018. doi: 10.5772/65599.
- [13] Z. Sun, *Cooperative Coordination and Formation Control for Multi-agent Systems*. Springer, 2018. doi : 10.1007/978-3-319-74265-6
- [14] S. Yang, J.-X. Xu, X. Li, D. Shen, *Iterative Learning Control for Multi-agent Systems Coordination*. Wiley-IEEE Press, 2017.
- [15] P. Scerri, R. Vincent, R.T. Mailler, *Coordination of Large-Scale Multiagent Systems*. Springer, 2010. doi : 10.1007/0-387-27972-5

- [16] O. Perminova-Harikovski, M. Gustafsson, K. Wikstrom, Defining Uncertainty in Projects – A New Perspective, volume 26(1) of *International Journal of Project Management*, 2008, pp. 73-79. doi: 10.1016/j.ijproman.2007.08.005.
- [17] E.Z.H. Zheng, M.M. Carvalho, Managing Uncertainty in Projects: A Review, Trends and Gaps, vol. 7(2) of *Revista de Gestao e Projetos GeP. Journal of Business and Projects*, 2016, pp. 95-99.
- [18] D. Cleden, *Managing Project Uncertainty (Advances in Project Management)*. 1st Edition. Routledge, 2017.
- [19] K. Macedo, M. Marinho, S. Santos, Uncertainty Management in Software Projects: A Case Study in a Public Company. *Journal of Convergence Information Technology* 14(1)(2019) 61-67.
- [20] V. Bryl, P. Giorgiani, S. Fante, ToothAgent: A Multi-agent System for Virtual Communities Support. In: Kolp M., Henderson-Sellers B., Mouratidis H., Garcia A., Ghose A.K., Bresciani P. (eds) *Agent-Oriented Information Systems IV. AOIS 2006*, volume 4898 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2008, pp. 212-230. doi: [https://doi.org/10.1007/978-3-540-77990-2\\_13](https://doi.org/10.1007/978-3-540-77990-2_13).
- [21] M. Fahad, O. Boissier, P. Maret, N. Moalla, C. Gravier, Smart Places: Multi-Agent based Smart Mobile Virtual Community Management System, volume 41 (4) of *Applied Intelligence*, Springer Verlag, Germany, 2014, pp. 1024-1042. doi: 10.1007/s10489-014-0569-2. hal-01015456.
- [22] Y. Lee, Q. Chong, Multi-agent Systems Support for Community-Based Learning, volume 15(1) of *Interacting with Computers*, 2003, pp. 33-55. doi: [https://doi.org/10.1016/S0953-5438\(02\)00057-7](https://doi.org/10.1016/S0953-5438(02)00057-7).
- [23] M. Hidalgo-Herrero, P. Rabanal, I. Rodriguez, F. Rubio, Comparing problem solving strategies for NP-hard optimization problems, vol. 124 (1-2) of *Fundamenta Informaticae*, 2013, pp. 1-25.
- [24] S.M. Abdulrahman, Using Swarm Intelligence for Solving NP-hard Problems, volume 6 (3) of *Academic Journal of Nawroz University*, 2017, pp. 46-50.
- [25] X. Huang, A polynomial-time algorithm for solving NP-hard problems in practice, volume 34 (1) of *ACM SIGACT News*, 2003, pp. 101-108.
- [26] B. Reus, How to Solve NP-Complete Problems. In: *Limits of Computation. Undergraduate Topics in Computer Science*, Springer, Cham, 2016, pp. 275-297. doi: [https://doi.org/10.1007/978-3-319-27889-6\\_21](https://doi.org/10.1007/978-3-319-27889-6_21).
- [27] G. Panchal, D. Panchal, Solving NP hard Problems using Genetic Algorithm, volume 6 (2) of *International Journal of Computer Science and Information Technology*, 2015, pp. 1824-1827.
- [28] M. Prates, P.H.C. Avelar, H. Lemos, L.C. Lamb, M.Y. Vardi, Learning to solve NP-complete problems : A graph neural network for decision TSP. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2019, pp. 4731-4738. doi : 10.1609/aaai.v33i01.33014731
- [29] L. Chyrun, E. Leshchynskyy, V. Lytvyn, A. Rzhеuskyi, V. Vysotska, Y. Borzov, Intellectual analysis of making decisions tree in information systems of screening observation for immunological patients, *CEUR Workshop Proceedings* 2488 (2019) 281–296.
- [30] M. Ummels, *Stochastic Multiplayer Games: Theory and Algorithms*. Amsterdam University Press, 2014.
- [31] V. Ungureanu, *Pareto-Nash-Stackelberg Game and Control Theory: Intelligent Paradigms and Applications*, Springer, 2018. doi : 10.1007/978-3-319-75151-1
- [32] S. K. Neogy, R. B. Bapat, D. Dubey, *Mathematical Programming and Game Theory*. Springer, 2018. doi : 10.1007/978-981-13-3059-9
- [33] A. Rzhеuskyi, O. Kutyuk, V. Vysotska, Y. Burov, V. Lytvyn, L. Chyrun, The Architecture of Distant Competencies Analyzing System for IT Recruitment, in: *Proceedings of the 14th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2019*, 2019, pp. 254–261.
- [34] H. Kushner, G. G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer Science & Business Media. 1997. doi : 10.1007/978-1-4899-2696-8
- [35] A. Benveniste, M. Metivier, P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag Berlin Heidelberg, 1990. doi : 10.1007/978-3-642-75894-2
- [36] V. Lytvyn, V. Vysotska, V. Mykhailyshyn, A. Rzhеuskyi, S. Semianchuk, System Development for Video Stream Data Analyzing. *Advances in Intelligent Systems and Computing* 1020 (2020) 315–331.