ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 4, No. 2, 2019

DISTRIBUTED ADAPTATION OF THE FUNCTIONAL INTEGRATION STRUCTURE OF A MULTI-AGENT SYSTEM IN A DUAL-TASKING ENVIRONMENT

Oleksii Botchkaryov

Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine
Author's e-mail: *alb2@ukr.net*

Submitted on 01.12.2019

***Abstract*: The paper considers the problem of distributed adaptation of the functional integration structure of a multi-agent system in a dual-tasking environment from the point of view of organizing multi-agent search and use of the functional emergence effect provided by different structures of functional integration. The considered problem belongs to a wider class of problems of structural adaptation and self-organization. Models of functional integration, in particular, models based on general quantitative characteristics of the functional roles distribution of agents and models based on local qualitative characteristics of the functional roles distribution of agents, taking into account the specifics of functional links established between agents have been considered in the paper. The problems of the distributed adaptation of the functional integration structure have been analyzed, including the problem of the functional specialization of agents in a multitasking environment. Various ways of organizing structural changes have been considered, including multi-agent parametric adaptation based on a local structural parameter. Multi-agent structural adaptation based on reinforcement learning methods, in particular, multi-agent structural adaptation based on the normalized exponential function method (MSA-softmax) and multi-agent structural adaptation based on the upper confidence bound method (MSA-UCB) has been proposed. The distributed adaptation methods simulation results have been presented, which showed the advantage of multi-agent structural adaptation over multi-agent parametric adaptation.**

Index Terms: structural adaptation, functional integration, multi-agent system

I. INTRODUCTION

The modern development and widespread adoption of mobile computing and wireless technologies determine the rapid growth of opportunities for solving the problems of distributed artificial intelligence, in particular the research and development of intelligent agents and multi-agent systems [1–4]. The relatively high performance of modern computing nodes, their large number in the distributed computer systems and networks, open up new possibilities for studying the concept of decentralized control in the context of multi-agent system technologies. Using decentralized control allows one to create multi-agent systems (MAS) with high survivability, reliability and wide scalability, and at the same time they are not inferior in efficiency to systems with centralized control, and in some classes of tasks they even outperform them.

One of the important areas of research in multi-agent systems technologies and decentralized control methods is the organization of multi-agent systems in multitasking environments when agents need to solve several tasks simultaneously. Examples of problems with a multitasking environment include the problem of multi-task learning [5–9], the problem of multi-task optimization [10–13], the problem of scheduling and task assignment in multi-agent systems [14–16], the problem of organizing distributed autonomous information gathering by a multi-agent system [17–21], etc. In problems with a multi-tasking environment, various options for the dependence between tasks are considered, starting from a set of independent tasks and ending with closely related tasks, for example, under the use of functional decomposition, where the overall task is broken down into a number of subtasks to form a hierarchical structure of dependencies. Wherein, the simplest for analysis, but nevertheless having a wide field of application, it is a model of a dual-tasking environment in which agents face two different tasks that are related to one degree or another.

The behavior of a multi-agent system in a multi-tasking environment implies a certain distribution of agents between the tasks to be solved. The local solution by the agent of a separate task is equivalent to accepting by the agent the corresponding function as a part of a multi-agent system. Performing this function for some time, the agent makes decisions and acts according to the selected functional role (mode of operation). Interaction of agents within the framework of their functions forms functional links between them. Let us define the functional integration structure (FIS) of a multi-agent system as the distribution of agents by functional roles and all corresponding functional links between them. In this case, of particular interest are situations where the actions of agents within the framework of their chosen functional roles are mutually complementary, giving the effect of functional emergence. That is, situations when, due to functional integration, a multi-agent system has fundamentally new qualities (capabilities) that are not available for its individual parts (agents). The use of new qualities (capabilities) obtained as a result of functional emergence allows the multi-agent system to solve tasks with greater efficiency, as well as to solve problems that cannot be solved by an individual agent. Thus, we can consider the following problem. How agents should search for and use the functional emergence effect, which is provided by different functional integration structures of a multi-agent system? To solve this problem, it is necessary to organize a collective search for such structures of functional integration that provide new qualities (capabilities) necessary for solving the tasks and (or) allowing them to be solved with greater efficiency. In other words, it is necessary to solve the problem of distributed adaptation of the functional integration structure of a multi-agent system, when each of the agents makes decisions on a local structure change on their own. Note that this task belongs to a wider class of problems of structural adaptation and self-organization [22–24]. We also note that in [20] an approach to solving this problem was proposed for the case of organizing the collective behavior of autonomous agents gathering information based on the corresponding model of a dual-tasking environment.

In the paper we propose models of functional integration, in particular, models based on general quantitative characteristics of the functional role distribution of agents and models based on local qualitative characteristics of the functional role distribution of agents, taking into account the specifics of functional links established between agents. The problems of the distributed adaptation of the functional integration structure are analyzed, including the problem of the functional specialization of agents in a multitasking environment. Various ways of organizing structural changes are considered, including multi-agent parametric adaptation based on a local structural parameter. We propose the multi-agent structural adaptation based on reinforcement learning methods, in particular, multi-agent structural adaptation based on the normalized exponential function method (MSA-softmax) and multi-agent structural adaptation based on the upper confidence bound method (MSA-UCB). The simulation results of the distributed adaptation methods are presented, which showed the advantage of multi-agent structural adaptation over multi-agent parametric adaptation.

II. MULTI-AGENT SYSTEM IN A DUAL-TASKING ENVIRONMENT

Consider a dual-tasking environment E(**T**), in which the general task **T**=$\{T_x,T_v\}$ is divided into two related sub-tasks T_x and T_v, each of which can be solved locally by a separate agent. In this case, the relationship between T_x and T_v determines the nature of the complementary relationship between the agents' actions of solving task **T**. To solve the T_x task, an agent can use the set of D_x actions, and to solve the T_v task, an agent can use the set of D_y actions. In terms of possibly overlapping D_x and D_y action sets, three options are possible:

1) $D_x \cap D_v = \varnothing$, D_x and D_v do not overlap;

2) $D_x \cap D_v = D_{xy}$, $D_{xy} \subset D_x$, $D_{xy} \subset D_y$, D_x and D_y overlap partially;

3) $D_x = D_v = D$, overlap completely.

To simplify the analysis, in this paper we will consider only the third option, when the same set of actions D is used to solve both tasks.

Based on the concept of decentralized control, we assume that each agent a_i makes a decision on the choice of a functional role and actions within this role independently of other agents. At each time step t, each agent a_i, i = 1, ..., N selects and implements the action d_t from the set of actions D. After that, each agent receives a local response from the environment to its action, based on which it can judge the success of the local solution of T_x and T_v tasks. From the perspective of the decision-making process, the agent can choose the action d_t within the framework of a local solution to either the T_x task or the T_v task. That is, the agent at each moment of time t chooses in the interests of which of the two tasks (T_x or T_v) to decide on the choice of action. Let us say that, choosing the action to solve the T_x task, the agent assumes the functional role A_x (operates in A_x mode), and choosing the action to solve the T_v task, the agent assumes the functional role A_v (operates in A_v mode). Thus, at each moment of time t, all agents $\{a_i\}$ are distributed into two functional roles (operating modes) A_x and A_v. Thereby, $n(A_x)+n(A_v)=N$, where $n(A_x)$ is the number of agents who have chosen the functional role A_x, $n(A_v)$ is the number of agents who have chosen the functional role A_y. The distribution of agents according to the functional roles *P(a,t)* defines the current functional integration structure of the multi-agent system, which is characterized by the corresponding effect of functional emergence. Moreover, different distributions *P(a,t)* (i.e., different functional integration structures) give a different effect of functional emergence. The multi-agent system is faced with the task of adapting the FIS to a previously unknown optimal structure of the functional decomposition of the **T** task into two sub-tasks T_x and T_v, that is, to find an FIS that gives the maximum effect of functional emergence in strength.

III. MODELS OF FUNCTIONAL INTEGRATION

From the point of view of methods for determining the effect of functional emergence, models of functional integration can be divided into the following two types.

1) Λ_G – models based on general quantitative characteristics of the distribution of agents according to functional roles; in the framework of these models, the strength indicator of the effect of functional emergence may, for example, depends on the ratio of the number of agents choosing one or another functional role: $E_G=F(n(A_x),n(A_y))$;

2) Λ_L – models based on local qualitative characteristics of the distribution of agents according to functional roles, taking into account the specifics of functional links established between agents; in the framework of these models, the indicator of the strength of the effect of functional emergence can, for example, be determined by comparing the functional roles in all N/2 pairs of agents (a_i,a_j) formed randomly: $E_L=F(\{A(a_i),A(a_j)\})$, where $A(a_i)$ is the functional role chosen by agent a_i.

In the case of a combination of these two types of models, the indicator of the strength of the functional emergence effect can be defined as $E = \Psi(E_G,E_L)$, where Ψ can be, for example, a weighted sum: $E=w_GE_G+w_LE_L$, $w_G + w_L = 1$.

Functional integration models Λ_L can be further divided into two classes of models according to the method of matching the functional roles of agents:

1) $\Lambda_L(\eta)$ – models based on random partitions of the set of agents into subsets of $m<N$ agents ($m = 2,3, ...$);

2) $\Lambda_L(\gamma)$ – models based on subgraphs of functional links between agents with $m<N$ vertices ($m = 2,3, ...$) of the complete graph of functional links.

In models of functional integration of type Λ_L let us define a component of coherence (a component of functional integration) as a subset of agents C{a} for which the effect of functional emergence is locally determined. In the case of random partitions (models $\Lambda_L(\eta)$), the component of coherence can be a pair of agents (a_i,a_j), a triplet of agents (a_i,a_j,a_h), etc. In the case of subgraphs of functional links (models $\Lambda_L(\gamma)$), the component of coherence can be a subgraph with two vertices a_i and a_j, a subgraph with three vertices a_i,a_j,a_h, etc. Let us define the matching scale L as the number of agents in the component of coherence. From the point of view of using different scales of matching, the functional integration model can be divided on the:

1) models with one fixed matching scale $L = 2,3,...,N$;

2) models with the simultaneous use of all matching scales in the range from 2 to $2 < L \leq N$;

3) models with a selective set of matching scales (for example, $L=\{2,4,5\}$).

Consider the model $\Lambda_L(\gamma,L)$ with a fixed matching scale L. In this model, for each component of coherence (a subgraph of L vertices) C{a}, a local indicator of the strength of the functional emergence effect $e_L(C\{a\})$ is determined. After that, the local results are combined into an overall result for the entire multi-agent system $E_L=F(\{e_L(C\{a\})\})$. In the simple case, the functional F can be a sum or an average. In a complex case, the functional F can be nonlinear, thereby reflecting the influence of the dependencies between the individual components of coherence on the overall indicator of the strength of the functional emergence effect. From the perspective of decentralized control, one can also distinguish between different options for informing an individual agent about the results of functional integration. For example, agents may only know an inaccurate value of the overall strength indicator of the obtained effect of functional emergence E_L, different from its true value. Or, agents can only know local indicators of the strength of the functional emergence effect $e_L(C\{a\})$ for those components of coherence C{a} where they participate.

In addition to the matching scale L in the model $\Lambda_L(\gamma,L)$, one can consider the degree of locality k, limiting the number of components of coherence (functional links subgraphs) C{a} with a given number of agent's neighbors $k<N$ in the graph of functional links. For example, for the matching scale $L=2$ and a connected regular graph of functional links, the maximum value of k is $k_{max}=N-1$ (complete graph) and the minimum value of k is $k_{min}=2$ (a cycle graph of length N). Let us consider several examples of the model $\Lambda_L(\gamma,L,k)$ with a connected regular graph of functional links. Example 1: $\Lambda_L(\gamma,L=2,k=2)$ – each agent has two functional links, the number of components of coherence is $n(C\{a\})=N-1$. Example 2: $\Lambda_L(\gamma,L=2,k=4)$ – each agent has four functional links, the number of components of coherence is $n(C\{a\})=2N-2$. Example 3: $\Lambda_L(\gamma,L=\{2,3\},k=4)$ – each agent participates in six components of coherence (four binary functional links and two triangle graphs of functional links), the number of components of coherence $n(C\{a\})=3N-1$. Note that in the models of functional integration $\Lambda_L(\gamma)$, the connectivity of the graph of functional links and its other parameters (regularity, diameter, etc.) are of great importance. In the basic version of the $\Lambda_L(\gamma)$ models, it is assumed that the graph of functional links is connected and regular.

From the point of view of the types of functional links in a dual-tasking environment, in the model $\Lambda_L(\gamma,L=2)$, three variants of functional integration are possible in a pair of agents: A_x~A_x, A_v~A_v, A_x~A_v (taking into account the equality A_x~A_v=A_v~A_x). That is, the component of coherence in the model $\Lambda_L(\gamma,L=2)$ can take three different "values" $C\{a\}\in\{A_x$~A_x, A_v~A_v, A_x~$A_v\}$. In the functional integration model $\Lambda_L(\gamma,L=3)$, four variants of the "values" of the component of coherence are possible: $C\{a\}\in\{A_x$~A_x~A_x, A_x~A_v~A_v, A_x~A_x~A_v, A_v~A_v~$A_v\}$ (taking into account the equality of the component-triangles with two identical vertices). One of the options for expanding the model $\Lambda_L(\gamma,L)$ can be the representation of the problem of choosing the "value" of the component of coherence C{a} as a problem of reinforcement learning in a stationary random environment (a multi-armed bandit problem) [25]. Then, for example, in $\Lambda_L(\gamma,L=2)$, for each component of coherence, three "actions" $\{A_x$~A_x, A_v~A_v, A_x~$A_v\}$ will be available, in response to which the corresponding pair of agents will receive a "reinforcement" as a local indicator of the strength of the effect of functional emergence $e_L(C\{a\})$. In total, in the model $\Lambda_L(\gamma,L)$ there will be $n(C\{a\})$ stationary random environments (multi-armed bandit problems) in the number of components of coherence.

In a dual-tasking environment with the model $\Lambda_L(\gamma,L=2)$, as a result of local decisions of agents about choosing a functional role we get three subgraphs of the graph of functional links, where each subgraph is formed by functional links of one corresponding type: $G(A_x{\sim}A_x)$, $G(A_y{\sim}A_y)$, $G(A_x{\sim}A_y)$. For each of these subgraphs, their general parameters (connectivity, diameter, etc.) can be determined. Thus, there is an interesting opportunity to analyze the functional integration structure based on a comparison of the parameters of these subgraphs. For example, a comparison of the diameters of the subgraphs $G(A_x{\sim}A_x)$, $G(A_y{\sim}A_y)$, $G(A_x{\sim}A_y)$ (or the diameters of their largest connected components) can show which type of functional link spans the most of the coordination space of a multi-agent system.

IV. THE PROBLEMS OF DISTRIBUTED ADAPTATION OF FIS

The central problem of the distributed adaptation of the FIS is the problem of organizing local search by an agent for a functional role within the framework of a coordinated multi-agent search for the best distribution of functional roles. By their local decisions on choosing a functional role, agents form an overall FIS of a multi-agent system, overcoming two uncertainties: 1) the uncertainty regarding the optimal structure of the functional decomposition of task **T** into two sub-tasks T_x and T_y, which corresponds to the desired FIS, which gives the maximum effect of functional emergence; 2) the uncertainty regarding the actions of other agents in choosing functional roles due to the decentralized control of multi-agent search for the best FIS. Additionally, distributed adaptation of FIS is characterized by the following problems.

1. The problem of choosing one action d_t to solve both tasks T_x and T_y simultaneously (for the case of completely or partially overlapping sets of actions D_x and D_y). A simple scenario, when the agent first chooses the functional role $A(a_i)$, and then chooses the action d_t within this role, can be complicated by the choice of the "compromise" action d_t, which to some extent would correspond to the decision logic of both functional roles A_x and A_y. That is, the agent needs to choose an action that would be simultaneously suitable for solving both tasks T_x and T_y. This problem has much in common with the problems of multi-objective optimization. The multi-agent parametric adaptation of FIS considered below can be regarded as one of the solutions to this problem.

2. The problem of functional specialization in a multitasking environment. Functional specialization refers to a situation where an agent spends most of his time solving one of the tasks. The result of this may be that the agent will solve this task better than others, for example, due to the accumulation of more experience in solving it in comparison with other tasks. This situation, in particular, occurs in problems solved by reinforcement learning methods. Thus, the agent's choice of a functional role is complicated by the need to additionally take into account how much of the time the agent has already spent on fulfilling one or another functional role, and how this affected the effectiveness of his work in these roles, taking into account the factor of functional specialization. For example, it is not advantageous for an agent to switch from a functional role in which he has achieved high performance indicators to a "new" functional role, which he rarely did before. Multitasking environments can differ in varying degrees of influence of the factor of functional specialization, starting from environments where this factor is absent and ending with environments where it plays a decisive role in solving problems. In addition, within the framework of a separate multitasking environment, the influence of the functional specialization factor may be different for different tasks. From the perspective of interaction between agents, functional specialization may be due to the following reasons: 1) the experience of solving a problem with this agent in its local environment is not suitable for an agent solving the same problem in another local environment (for example, agents solve multi-armed bandit problems with different payoff functions); 2) the rate of transfer of experience in solving a problem from one agent to another is too low compared to the rate of change of environmental parameters (that is, the transferred experience becomes outdated until it is transferred to another agent).

3. The problem of choosing the degree of functional specialization by an agent. In multitasking environments with a strong influence of the factor of functional specialization, an agent faces the problem of distributing his time between solving different tasks (that is, performing different functional roles), taking into account the fact that the less time is spent on solving a task, the lower is the effectiveness of solving it. For example, in a dual-tasking environment, an agent has two alternatives: 1) spend approximately the same time on solving tasks T_x and T_y, without specializing in solving any of them; 2) spend more time solving one of the tasks, specializing in it. The first alternative is preferable in situations with greater uncertainty regarding the optimal FIS and the corresponding functional role of the agent, and the second alternative is preferable in the opposite case. In other words, it is not profitable for an agent to specialize in solving a task until it becomes clear that the local choice of the corresponding functional role provides the desired optimal distribution $P(a,t)^*$.

4. The problem of the distribution of agents by functional roles, taking into account the performance values they achieved in solving the corresponding tasks. Assume that, within the framework of the functional integration model Λ_G, the desired ratio $n(A_x)/n(A_y)$ corresponding to the FIS with the maximum effect of functional emergence is found. Then it is beneficial that among the $n(A_x)$ agents with the functional role A_x there are those agents who cope with the solution of the task T_x better than all other agents (for example, as a result of functional specialization in solving this task). And the same is true for the functional role A_y. That is, the agent,

when choosing a functional role, must additionally take into account how much better or worse than other agents it copes with it. In other words, for a given best distribution of functional roles *P(a,t)**, a role A_i should go to those agents who have achieved the best performance values for solving the corresponding task.

V. METHODS OF ORGANIZING STRUCTURAL CHANGES

For distributed adaptation of the structure of a multi-agent system, different methods of organizing effective structural changes can be used. Among them, two groups of methods can be distinguished:

1) M_p – methods based on the gradual change of one or more local structural parameters, which determine the functional role of an agent;

2) M_s – methods based on direct switching between functional roles of agents within a separate decision-making process.

The methods M_s have the following advantages.

1. They allow one to accumulate and use the experience of structural changes. Using the collective memory of structures, agents are given the opportunity to remember which structures have been successful in the past. Accordingly, when the situation changes, they rely on this experience and begin to search for a new structure with those options that were most successful in the past. Moreover, changes in the structure can be "fast" and large-scale, as opposed to "slow" and gradual changes made by the methods of the first group.

2. They can provide a synergistic effect from the use of several previously found structures suitable for different situations. By quickly switching and combining such structures, the expansion of adaptation capabilities is achieved due to the use of the beneficial properties of each structure. This allows a multi-agent system to obtain new properties that are not inherent in any of the structures individually.

The methods for organizing structural changes M_s include the choice of time when the agent switches from one functional role to another. In the basic version, such a switch occurs with some time period τ_s, during which the agent accumulates the information necessary to make a decision about switching to another functional role. One of alternatives is when the decision to switch the functional role is triggered by a certain event, for example, a decrease in the agent's performance indicator below a predetermined threshold value.

One of the most perspective approaches to the organization of effective structural changes in the decentralized control mode is the use of reinforcement learning methods [25–29]. Further, we assume that, in the framework of this approach, in a given dual-tasking environment each agent at time t receives the following information about the results of the action d_t: 1) $r_x(d)$ – the local reward (reinforcement) in the task T_x, which is calculated using u(d,t) – the estimation of efficiency of solving T_x by the joint actions of agents; 2) $r_y(d)$ – local reward (reinforcement) in the task Ty, which is calculated using v(d,t) – the estimation of efficiency of solving T_v by the joint actions of agents; 3) $r_s(d)$ – the reward obtained due to the effect of functional emergence provided by the current structure of the functional integration of the MAS.

As an example of methods M_p with the use of reinforcement learning, let us consider multi-agent parametric adaptation (MPA) of FIS based on the local structural parameter s(a). In this method, the overall reward r(d) is calculated as the weighted sum of the rewards in T_x and T_v: $r(d) = q_x r_x(d) + q_v r_y(d)$, where $q_x, q_v \in [0,1]$, $q_x + q_v = 1$. The local structural parameter $s(a) = q_x - q_v$, $s(a) \in [-1,+1]$ determines the ratio of the weights with which the agent's rewards in T_x and T_v are taken into account in the reinforcement learning method. If $s(a)>0$ ($q_x>q_v$), then when choosing the next action, the agent's reward in T_x is taken into account with greater weight, and accordingly, the agent is more likely to choose the action within the functional role A_x. If $s(a)<0$ ($q_x<q_v$), then the agent is more likely to choose an action within the functional role A_v. Thus, the agent's choice of a particular functional role depends on the value of the structural parameter s(a). The structural parameter s(a) is adjusted with a step $0 < \Delta < 1$ (for example, $\Delta = 0.001$) depending on $r_s(d)$. In the case of positive reinforcement $r_s(d)$, the parameter s(a) is changed so as to further increase the weight of the local reward corresponding to the current functional role. In the case of negative reinforcement $r_s(d)$, the parameter s(a) changes so as to shift the weight of the local reward in favor of a different functional role. Note that from the perspective of choosing the parameters of the reinforcement learning, two cases of multi-agent parametric adaptation can be distinguished: 1) MPA with local optimization over the set of actions and time steps (MPA-L) and 2) MPA with global optimization over actions and time steps (MPA-G).

VI. MULTI-AGENT STRUCTURAL ADAPTATION IN A DUAL-TASKING ENVIRONMENT

Consider multi-agent structural adaptation (MSA) of FIS in a dual-tasking environment as one of the M_s methods. In MSA, the agent's decision-making process is divided into two levels: 1) the structural level at which the agent selects one of two functional roles (operating modes) $A_i=\{A_x,A_y\}$, and 2) the functional level at which the agent selects action d in accordance with the selected functional role (operating mode) A_i.

Consider the procedure of MSA as a process of distributed adaptation of the structure of functional integration

$$S = < P(a,t),\ \{\sigma(a,a')\} >,$$

where *P*(*a*,*t*) is the distribution of agents by functional roles, $\{\sigma(a,a')\}$ is the set of coordination links between agents within the functional links defined by *P*(*a*,*t*).

Let us present the operating modes of an agent at the functional level as $A_x=(R_x(a),U_x(a))$ – decision-making on the choice of actions to solve the problem T_x; $A_y=(R_y(a),U_y(a))$ – decision-making on the choice of actions to solve the problem T_y; and at the structural level $A_s=(R_s(a),U_s(a))$ – making decisions about the actions to discrete change of structure S by switching between modes A_x and A_y. Then the multi-agent structural adaptation can be represented as

$$\mathrm{MSA} = < [A_x=(R_x(a),U_x(a)),\; A_y=(R_y(a),U_y(a))],\; A_s=(R_s(a),U_s(a)) >,$$

where $R_x(a)$: $r_x(d)=f_u(\{u(d,t)\})$ is the reward function of action d in operating mode A_x, $R_y(a)$: $r_y(d)=f_v(\{v(d,t)\})$ is the reward function of action d in operating mode A_y, $R_s(a)$: $r_s(g)=f_r(\{r_x(d)\},\{r_y(d)\})+F(\{r_s\}_{k(t)})$ is the reward function of choosing the operating mode (functional role), where $g\in\{A_x,\ A_y\}$, $f_r(\{r_x(d)\},\{r_y(d)\})$ is the function for evaluating performance of operating in modes A_x and A_y, $F(\{r_s\}_{k(t)})$ is a function of estimating the strength of the functional emergence effect in the components of coherence with the degree of locality k(t) = 0,…, N-1. *U(a)* is the procedure of decision making. In particular

$$U_x(a)=\{F_x(t), f_x(d_t,t), J_x(a)\},$$

$$U_y(a)=\{F_y(t), f_y(d_t,t), J_y(a)\},$$

where $F_x(t)$, $F_y(t)$ are the utility functions of the functional level, $f_x(d_t,t)$, $f_y(d_t,t)$ are the functions of switching between actions (in the corresponding operating modes A_x and A_y), J (a) is the method of information interaction and coordination with other agents:

$$J_x(a) = \{\sigma(A_x,A_x),\ \sigma(A_x,A_y)\},$$

$$J_y(a) = \{\sigma(A_y,A_y),\ \sigma(A_y,\ A_x)\}.$$

Functional level utility functions are as follows:

$$F_x(t) = \frac{1}{t}\sum_{i=1}^{t} r_{x,i}(d),$$

$$F_y(t) = \frac{1}{t}\sum_{i=1}^{t} r_{y,i}(d),$$

where t is the current time step, $r_{x,i}(d)$ is the reward that determines the performance of the agent in operating mode A_x at time step t, $r_{y,i}(d)$ is the reward that determines the performance of the agent in operating mode A_y at time step t. The utility function $F_x(t)$ reflects the overall performance of the agent in solving task T_x. The utility function $F_y(t)$ reflects the overall performance of the agent in solving task T_y.

The procedure of decision making at the structural level

$$U_s(a)=\{\ F_s(t), f_s(g_t,t), J_s(a),\ \tau_s\ \},$$

where $f_s(g_t,t)$ is the function of switching between functional roles (operating modes) (A_x,A_y), $J_s(a) = \{\sigma(A_s,A_s)\}$, τ_s is the time step of $U_s(a)$ activation, $F_s(t)$ is the utility function of the structural level:

$$F_s(t) = \frac{\tau_s}{t}\sum_{i=1}^{t/\tau_s} r_{s,i}(g).$$

In each of the two operating modes of the functional level (A_x and A_y), the procedure for optimizing the choice of actions is performed. At the same time, at the structural level, optimization of the distribution of agents by the functional roles (operating modes) A_x and A_y is performed. As an optimization procedure, methods of reinforcement learning in a stationary random environment are used, including 1) the method of normalized exponential function (softmax action selection) [25] and 2) the method of Upper-Confidence-Bound [30–31]. For each of these methods, an action value is calculated:

$$Q_{x,t}(d) = Q_{x,t-1}(d) + \alpha_x(r_{x,t}(d) - Q_{x,t-1}(d)),$$

$$Q_{y,t}(d) = Q_{y,t-1}(d) + \alpha_y(r_{y,t}(d) - Q_{y,t-1}(d)),$$

$$Q_{s,t}(g) = Q_{s,t-1}(g) + \alpha_s(r_{s,t}(g) - Q_{s,t-1}(g)),$$

where α_x, α_y, $\alpha_s \in(0,1]$ are the learning steps.

In a multi-agent structural adaptation based on normalized exponential function (MSA-softmax) (Fig. 1), the next action is selected with the probability:

$$p_{x,t}(d) = \frac{e^{Q_{x,t}(d)/\mu_x}}{\sum_{D_a} e^{Q_{x,t}(b)/\mu_x}},$$

$$p_{y,t}(d) = \frac{e^{Q_{y,t}(d)/\mu_y}}{\sum_{D_a} e^{Q_{y,t}(b)/\mu_y}},$$

$$p_{s,t}(g) = \frac{e^{Q_{s,t}(g)/\mu_s}}{\sum_{A_w,A_r} e^{Q_{s,t}(h)/\mu_s}},$$

where $Q_t(d)$ is the action value of d, D_a is a subset of actions which are currently available for an agent $(b\in D_a)$, μ is the scaling factor (μ > 0, μ=const).

In a multi-agent structural adaptation based on Upper-Confidence-Bound (MSA-UCB) (Fig. 2), the next action is selected as:

$$d_{t+1} = \mathrm{argmax}_{D_a}\left(Q_{x,t}(d) + c_x\sqrt{\frac{\ln(t)}{k_{x,t}(d)}}\right),$$

$$d_{t+1} = \mathrm{argmax}_{D_a}\left(Q_{y,t}(d) + c_y\sqrt{\frac{\ln(t)}{k_{y,t}(d)}}\right),$$

$$g_{t/\tau_s+1} = \mathrm{argmax}_{\{A_x,A_y\}}\left(Q_{s,t}(d) + c_s\sqrt{\frac{\ln(t/\tau_s)}{k_{s,t/\tau_s}(d)}}\right),$$

where c is the scaling factor (c > 0).

When the agent operates in the mode A_x, learning in the framework of the functional role A_y is performed in passive mode, and vice versa, when the agent operates in the mode A_y, learning in the framework of the functional role A_x is performed in passive mode.

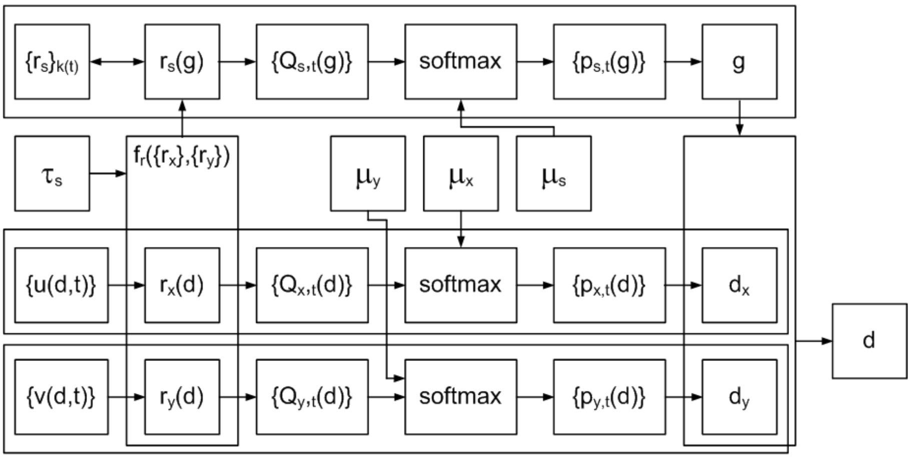

Fig. 1. The outline of MSA based on the normalized exponential function (MSA-softmax)

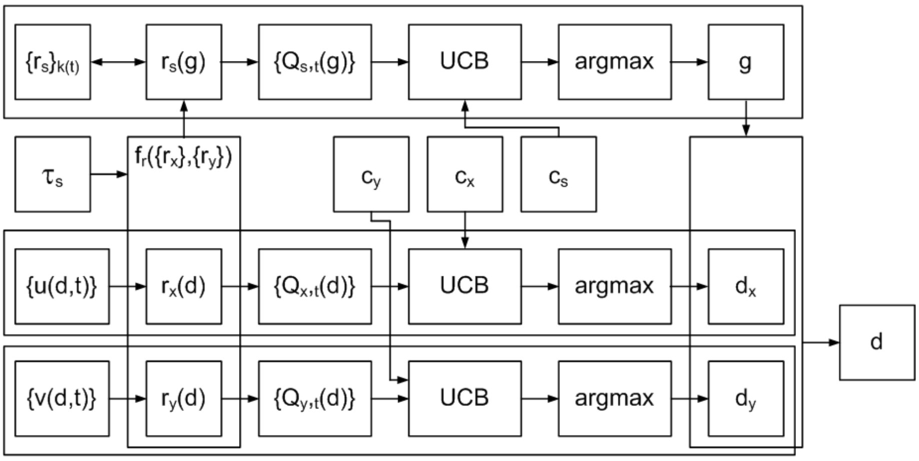

Fig. 2. The outline of MSA based on the upper confidence bound (MSA-UCB)

VII. SIMULATION RESULTS

The presented above methods of the distributed adaptation (MSA and MPA) in a dual-tasking environment $E(\mathbf{T})=\{E(T_x), E(T_y), \mathbf{T}=\{T_x,T_y\}\}$ were studied for the case of a functional integration model Λ_G and completely overlapping sets of action $D_x = D_y = D$ (with M denoting the number of actions). For the purpose of simulation the $E(T_x)=\{MAB_{ul}, MAB_{ug}, R_x(d), \lambda\}$ is modeled as a composed multi-armed bandit problem with a reward function $R_x(d)$ and $E(T_y)=\{G_r,R_y(k)\}$ is modeled as a homogeneous symmetric game [32–34] with a reward function $R_y(k)$. The reward function $R_x(d)$ is determined by the reward probabilities $\{p_r(d)\}_N$ such as

$$R_x(d):\ p_r(d) = \lambda \times p_{u,g}(d) + (1 - \lambda) \times p_{u,l}(d),$$

where $p_{u,l}(d)$ is the reward probability in a multi-armed bandit problem $MAB_{ul}=\{p_{u,l}(d)\}$, $p_{u,g}(d)$ is the reward probability of a multi-armed bandit problem $MAB_{ug}=\{p_{u,g}(d)\}$, λ is the weight coefficient that determines the degree of dependence between reward function $R_x(d)$ and reward probabilities in random environments MAB_{ul} and MAB_{ug}.

The composed multi-armed bandit problem (stationary random environment) $E(T_x)$ is used to model the operation of agents in the functional role (mode) A_x. The $E(T_x)$ is presented as a combination of two multi-armed bandit problems (random environments): MAB_{ul}, which simulates the local component of agents' behavior without information about functional links between agents, and MAB_{ug}, which simulates the global component of the agents' behavior based on information about functional links between agents. The weight coefficient λ determines the degree to which the reward function $R_x(d)$ depends on the rewards of the local and global components of agents' behavior. The homogeneous symmetric game $E(T_y)$ is used to model the operation of agents in the functional role (mode) A_y. The relationship between tasks T_x and T_y is modeled in the following way. If at time step *t* the collective actions of agents in a homogeneous symmetric game $E(T_y)$ give a win, then in a random environment $E(T_x)$ agents are awarded local rewards with probability $p_r(d)$, otherwise in the case of a loss in $E(T_y)$ in a random environment $E(T_x)$ agents are awarded local rewards with probability $p_{u,l}(d)$. In this way we study adaptive capabilities of agents depending on the presence or absence of information about the functional links between agents due to their functional roles.

In a homogeneous symmetric game $E(T_y)=\{G_r,R_y(k)\}$, the agent's choice of action d with some parameter $h_i(d)$ is equivalent to the choice of strategy $h_i \in H_g$. In the simplest case, $H_g=\{h_1, h_2\}$ and the reward $r \in \{0;1\}$. Thus the game G_r is defined by the reward function $R_y(k)$ $(0 < k < 1, 0 < R_g(k) < 1)$, where $R_y(k)$ is the mathematical expectation of the reward of each player (agent) in a game instance, in which kN players chose strategy h_1 and (1–k)N players chose strategy h_2. Function $R_y(k)$ was chosen as a piecewise constant function [32, 33]: 1) the value of k is divided into equal intervals $K_i=(k_i,k_{i+1})$, i=1,…,n; 2) each interval has its own mathematical expectation of reward $R_y(K_i)$. At each step t, one instance of the game G_r is played and each player (agent), depending on his choice of d, gets reward $r_{y,i}(d) \in \{0;1\}$.

The MSA performance (separately for the MSA-softmax and for the MSA-UCB) and MPA (separately for the MPA-L and for the MPA-G) was evaluated as

$$w(Q) = Q / Q_{max} ,$$

where Q is the total reward in a dual-tasking environment E(**T**) (time-averaged and averaged over all agents):

$$Q=\frac{1}{N}\sum_{A}\left[\frac{1}{T}\sum_{i=1}^{T}(r_{x,i}(d)+r_{y,i}(d))\right],$$

Q_{max} is the maximum possible total reward in a dual-tasking environment E(**T**) (time-averaged and averaged over all agents).

During the simulation, we studied the work of four methods of distributed adaptation (Fig. 3–6): 1) multi-agent structural adaptation based on a normalized exponential function (MSA-softmax), 2) multi-agent structural adaptation based on upper confidence bound (MSA-UCB), 3) multi-agent parametric adaptation with local optimization (MPA-L), 4) multi-agent parametric adaptation with global optimization (MPA-G). Simulation of these adaptation methods (Fig. 3–6) showed the advantage of multi-agent structural adaptation over multi-agent parametric adaptation. In addition, the advantage of MSA over MPA increases with the increase of dependence between reward in the random environment MAB_{ug} and reward function $R_s(d)$ in the random environment $E(T_x)$ (value λ), i.e., with an increase of the weight of the global component in agents' behavior based on information about functional links between agents. The simulation parameters (Fig. 3–6) are as follows: the number of time steps in one simulation *T* = 1000, the number of repetitions of simulations n = 10000, c = 2.0 (MSA-UCB), μ= 0.05 (MSA-softmax), τ_s = 10 (MSA), N = {5-50}, M = 4. On average, for various combinations of parameters of adaptation methods and simulation parameters (M = {2, ..., 20}, N = {5 , ..., 100}, λ = {0.6, 0.7, 0.8, 0.9}) the advantage of MSA over MPA is 23.2 %.

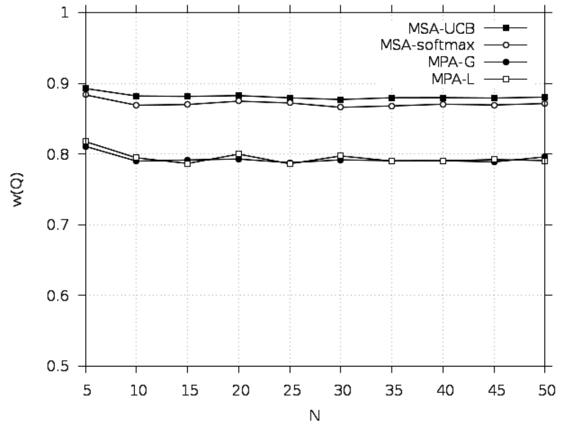

Fig. 3. Simulation results: λ = 0.6, M = 4, T = 1000, n = 10000

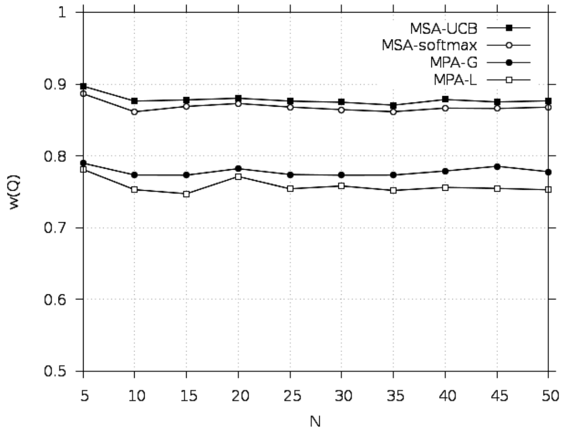

Fig. 4. Simulation results: λ=0.7, M=4, T=1000, n=10000

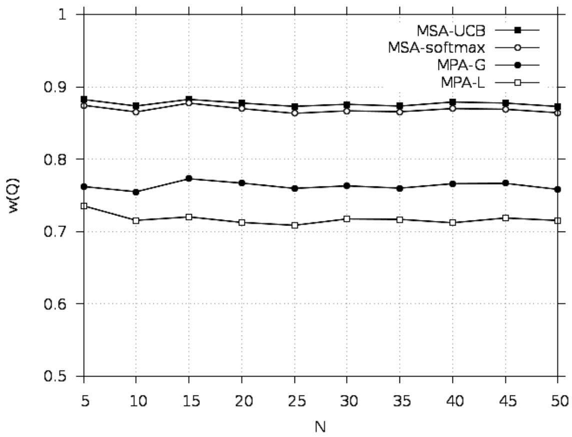

Fig. 5. Simulation results: λ=0.8, M=4, T=1000, n=10000

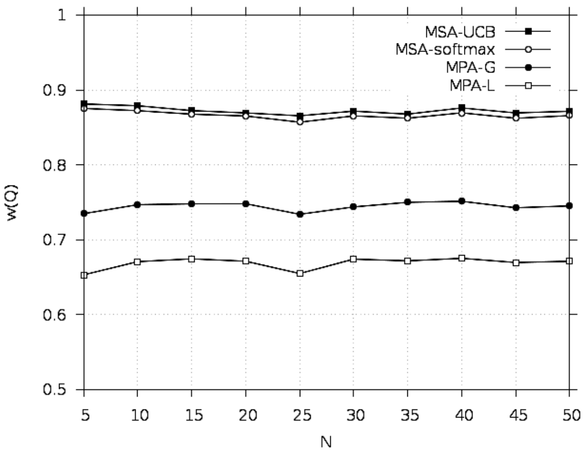

Fig. 6. Simulation results: λ=0.9, M=4, T=1000, n=10000

VIII. CONCLUSIONS

The problem of distributed adaptation of the functional integration structure of a multi-agent system in a dual-tasking environment from the point of view of organizing multi-agent search and use of the functional emergence effect provided by different structures of functional integration were considered. The considered problem belongs to a wider class of problems of structural adaptation and self-organization. The models of functional integration, in particular, models based on general quantitative characteristics of the functional role distribution of agents and models based on local qualitative characteristics of the functional role distribution of agents, taking into account the specifics of functional links established between agents were considered in the paper. The problems of the distributed adaptation of the functional integration structure were analyzed, including the problem of the functional specialization of agents in a multitasking environment. Various ways of organizing structural changes were considered, including multi-agent parametric adaptation based on a local structural parameter. The multi-agent structural adaptation based on reinforcement learning methods, in particular, multi-agent structural adaptation based on the normalized exponential function method (MSA-softmax) and multi-agent structural adaptation based on the upper confidence bound method (MSA-UCB) were regarded. The simulation results of the distributed adaptation methods were presented, which showed the advantage of multi-agent structural adaptation over multi-agent parametric adaptation.

REFERENCES

[1] *Multiagent Systems*, by Gerhard Weiss (Editor), 2nd edition, The MIT Press, 2013. 920 p.

[2] Michael Wooldridge, An Introduction to MultiAgent Systems, 2nd edition, Wiley, 2009. 484 p.

[3] Yoav Shoham, Kevin Leyton-Brown, *Multiagent Systems: Algorithmic,Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2008. 504 p.

[4] *Multi-Agent Systems: Simulation and Applications*, by Adelinde M. Uhrmacher (Editor), Danny Weyns (Editor), CRC Press, 2009. 566 p.

[5] Caruana, R. (1997). *Multitask Learning. Machine Learning*, 28(1), 41–75

[6] Thung, K., Wee, C. *A brief review on multi-task learning*, Multimedia Tools and Applications, Vol. 77, 2018. pp. 29705–29725

[7] Zhang, Y., & Yeung, D.-Y. (2012, March 15). *A Convex Formulation for Learning Task Relationships in Multi-Task Learning*, in Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI 2010), Catalina Island, CA, USA, July 8–11, 2010. pp. 733–742

[8] Ando, R.; Zhang, T. (2005) *A framework for learning predictive structures from multiple tasks and unlabeled data*. The Journal of Machine Learning Research. 6: 1817–1853

[9] Argyriou, A., Evgeniou, T., Pontil, M. (2008) *Convex multi-task feature learning*. Machine Learning. 73 (3): 243–272

[10] Swersky, K., Snoek, J., & Adams, R. P. (2013) Multi-task bayesian optimization. Advances in neural information processing systems. pp. 2004–2012

[11] Ong, Y. S., & Gupta, A. (2016). Evolutionary multitasking: a computer science view of cognitive multitasking. Cognitive Computation, 8(2). pp. 125–142

[12] Cheng, M. Y., Gupta, A., Ong, Y. S., & Ni, Z. W. (2017). Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design. Engineering Applications of Artificial Intelligence, Vol. 64. pp. 13–24

[13] Bao, L., Qi, Y., Shen, M., Bu, X., Yu, J., Li, Q., & Chen, P. (2018, June). An Evolutionary Multitasking Algorithm for Cloud Computing Service Composition. In World Congress on Services. Springer, Cham. pp. 130–144

[14] Uchibe, E., Kato, T., Hosoda, K., & Asada, M. (2001). Dynamic task assignment in a multiagent/multitask environment based on module conflict resolution. Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, 4, vol. 4., 3987–3992

[15] Alessandro Agnetis, Jean-Charles Billaut, Stanisław Gawiejnowicz, Dario Pacciarelli, Ameur Soukhal, Multiagent scheduling: Models and algorithms, Springer; 2014. 271 p.

[16] Li, Shisheng & Chen, Ren-Xia & Tian, Ji. (2019). Multitasking scheduling problems with two competitive agents. Engineering Optimization. 1–17

[17] Botchkaryov A., Golembo V., Applying intelligent technologies of data collection to autonomous cyber-physical systems, Transactions on Computer systems and networks, Lviv Polytechnic National University Press, No. 830, 2015. pp. 7–11 (in Ukrainian)

[18] Melnyk A., Golembo V., Botchkaryov A., The new principles of designing configurable smart sensor networks based on intelligent agents, Transactions on Computer systems and networks, Lviv Polytechnic National University Press, No. 492, 2003. pp. 100–107 (in Ukrainian)

[19] Botchkaryov A., Collective behavior of mobile intelligent agents solving the autonomous distributed exploration task, Transactions on Computer systems and networks, Lviv Polytechnic National University Press, No. 546, 2005. pp. 12–17 (in Ukrainian)

[20] Botchkaryov A., Structural adaptation of the autonomous distributed sensing and computing systems, Transactions on Computer systems and networks, Lviv Polytechnic National University Press, No. 688, 2010. pp. 16–22 (in Ukrainian)

[21] Botchkaryov A., The problem of organizing adaptive sensing and computing processes in autonomous distributed systems, Transactions on Computer systems and networks, Lviv Polytechnic National University Press, No. 745, 2012. pp. 20–26 (in Ukrainian)

[22] Serge Kernbach, Structural Self-Organization in Multi-Agents and Multi-Robotic Systems, Logos Verlag, 2008. 250 p.

[23] Preisler, Thomas & Renz, Wolfgang. (2015). Structural Adaptations for Self-Organizing Multi-Agent Systems, The Seventh International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE 2015), At Nice, France

[24] Jiao, W., & Sun, Y. (2016). Self-adaptation of multi-agent systems in dynamic environments based on experience exchanges. Journal of Systems and Software, 122, 165–179

[25] Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, 2nd edition, The MIT Press, 2018. 552 p.

[26] L. P. Kaelbling, Michael L. Littman, and Andrew W. Moore, Reinforcement learning: A survey. Journal of AI Research, N 4, 1996. pp. 237–285

[27] Csaba Szepesvari, Algorithms for Reinforcement Learning, Morgan and Claypool Publishers, 2010. 104 p.

[28] Maxim Lapan, Deep Reinforcement Learning Hands-On, Packt Publishing, 2018. 546 p.

[29] Schwartz H. M. Multi-Agent Machine Learning: A Reinforcement Approach, Wiley, 2014. 256 p.

[30] Auer, P. Using upper confidence bounds for online learning. Proceedings 41st Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc., 2000. pp. 270–279

[31] Auer, P., Cesa-Bianchi, N., Fischer, P., Finite-time analysis of the multiarmed bandit problem. Machine learning, 47 (2–3), 2002. pp. 235–256

[32] Tsetlin, M. L., Automaton Theory and Modeling of Biological Systems. Academic Press, New York, 1973. 288 p.

[33] Varshavsky, V. I., Collective behavior of automata, Moscow, Nauka, 1973. 408 p.

[34] Narendra, K. and Thathachar, M. A. L., Learning Automata: An Introduction, 2nd ed., Dover Publications, 2013. 496 p.

Oleksii Botchkaryov was born in 1975 in Lviv, Ukraine. He received the B.S. and the M.S. degrees in Computer Engineering at Lviv Polytechnic National University in 1998 and the Ph.D. degree in Computer Systems and Components at Lviv Polytechnic National University in 2019. He has been doing scientific and research work since 1994. He is the author of a book (with coauthors) and more than 30 articles. Currently, he is a Senior Lecturer at the Computer Engineering Department and a Research Fellow at the Intelligent Systems Laboratory, Lviv Polytechnic National University. His research interests include self-organization in complex systems, structural adaptation, intelligent information-gathering agents and multi-agent systems.