ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 7, Num. 2, 2022

COSCHEDULING SPATIAL SELF-ORGANIZATION AND DISTRIBUTED DATA COLLECTION IN MULTI-AGENT SYSTEMS

Alexey Botchkaryov

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Author's e-mail: *oleksii.y.bochkarov@lpnu.ua*

https://doi.org/10.23939/acps2022.__.___
Submitted on 26.09.2022

***Abstract*: The problem of coscheduling spatial self-organization control processes and distributed data collection processes in a multi-agent system has been considered. The goal of coscheduling is to find and use the possibilities of functional coordination of these processes and increase the efficiency of the multi-agent system due to their parallel execution. An analysis of the main features of spatial self-organization tasks that affect the solution of the problem of coscheduling has been carried out. Variants of the mobile agent robotic platform configuration and the problem of the dependence of spatial self-organization algorithms on the type of robotic platform have been considered. A method of coscheduling of spatial self-organization and distributed data collection by coordinated parallel execution of the corresponding data collection process and the process of controlling mobile agent motion has been proposed. The method of coscheduling is implemented using the interaction protocol of these processes and the algorithm for planning their parallel execution using functional decomposition. The simulation results of the proposed method of coscheduling are given. It is proved that the proposed method of coscheduling provides acceleration of computations in the decision-making module of the mobile agent due to more efficient parallelization. On average, for typical values of parameters of control processes, the proposed method of coscheduling provides acceleration of computations in the decision-making module of the mobile agent by 40.6 %.**

***Index Terms:* coscheduling, distributed data collection, spatial self-organization, mobile agent, multi-agent system.**

I. INTRODUCTION

Currently, there is a wide variety of mobile robotics platforms that are used for building mobile agents, in particular mobile agents whose behavior is controlled by adaptive processes of distributed data collection. Based on mobile agents, distributed robotics systems and multi-agent systems [1] are built using decision-making methods in conditions of decentralized control [2] and methods for coordinating joint collective actions of agents [3]. An important class of such systems is formed by multi-agent systems that collect data on distributed control objects or distributed physical processes in cyber-physical systems [4], for example, on physical processes in the World Ocean [5]. The problem of distributed data collection has many aspects, in particular, the search for the optimal placement of sensor nodes in space [6], including the use of reinforcement learning methods [7] and self-organization principles [8]. In many cases, to solve the problems of distributed data collection, multi-agent systems need to solve the corresponding problems of spatial self-organization [9]. In the most general case, spatial self-organization is the formation and coordinated movement of mobile agents in some space under the conditions of decentralized control [10]. The tasks of spatial self-organization are solved by developing methods and algorithms for the collective behavior of mobile agents as parts of a multi-agent system [11]. In some cases, specialized solutions are developed for separate problems of spatial self-organization, for example, the problem of multi-agent coverage control [12], the problem of decentralized cluster formation [13], the problem of controlling a team of mobile agents for object transportation [14], etc. In most cases, problems of spatial self-organization are solved with various types of uncertainty [15]. Let us emphasize that the study and design of spatial self-organization methods is one of the main directions in the development of multi-agent systems technologies.

Since the multi-agent system simultaneously solves the tasks of distributed data collection and related tasks of spatial self-organization, there is a problem with the functional coordination of the corresponding processes of controlling the behavior of a mobile agent. This problem has several aspects, in particular the degree and nature of the functional dependence of the corresponding algorithms, the degree of their dependence on the features of the mobile robotics platform of the agent [16], the possibility and degree of parallelization of these algorithms, methods of coordinating their simultaneous execution, etc. One of the most promising approaches to solving this problem is the study of methods of coscheduling spatial self-organization control processes and distributed data collection processes. The goal of coscheduling is to identify and use the possibilities of functional coordination of these processes and increase the efficiency of the multi-agent system due to their parallel execution.

The article discusses the problem of coscheduling spatial self-organization control processes and distributed data collection processes in a multi-agent system. The purpose of coscheduling is to find and use the possibilities of functional coordination of these processes and increase the efficiency of the multi-agent system due to their parallel execution. A method of coscheduling of spatial self-organization and distributed data collection by coordinated parallel execution of the corresponding data collection process and the process of controlling mobile agent motion has been proposed. The simulation results of the proposed method of coscheduling are given. It is proved that the proposed method of coscheduling provides acceleration of computations in the decision-making module of the mobile agent due to more efficient parallelization.

II. OBJECTIVES

To solve the problem of coscheduling spatial self-organization control processes and distributed data collection processes in a multi-agent system, it is necessary to reach the following objectives:

1. Analyze the features of the spatial self-organization tasks that affect the solution of the problem of coscheduling.
2. Find the main types of uncertainty that make it difficult to solve problems of spatial self-organization and provide methods for evaluating the efficiency of spatial self-organization algorithms.
3. Consider the problem of different variants of the mobile agent robotic platform configuration and the problem of the dependence of spatial self-organization algorithms on the type of robotic platform.
4. Find approaches to the development of a set of spatial self-organization algorithms, independent of the robotic platform of a mobile agent.
5. Analyze the problem of coscheduling spatial self-organization and distributed data collection.
6. Develop a method for coscheduling spatial self-organization and distributed data collection based on the results of the analysis and taking into account the specifics of spatial self-organization tasks. The method should provide an acceleration of computations in the decision-making module of the mobile agent by at least 10 %.
7. Simulate the developed method of coscheduling and evaluate its efficiency.

III. MAIN FEATURES OF SPATIAL SELF-ORGANIZATION TASKS

The problem of self-organization of a multi-agent system in space arises in situations where there is uncertainty either about the space itself or about the actions of other agents of the multi-agent system [9-11]. Thus, the collective of mobile agents gradually overcomes this uncertainty in the process of spatial self-organization. Note that different problems of spatial self-organization can be classified based on the type of uncertainty that agents overcome when solving these problems. Depending on what exactly is unknown to agents, three main types of uncertainty can be distinguished, which can be present in one way or another in problems of spatial self-organization (Table 1).

The source of uncertainty U(S) is the spatial characteristics of the environment, in particular, the configuration of the space in which the multi-agent system is located (boundaries, terrain, dangerous areas, etc.). In the most complex cases, this configuration can change dynamically. In the case of uncertainty U(A), two main options can be distinguished: 1) agents cooperate; 2) agents counteract with each other. In the case of uncertainty U(P), it is assumed that the dynamic characteristics of the environment affect the movement of the mobile agent in some way (moving obstacles, currents, changes in the friction coefficient, etc.). This is especially important for mobile agents moving in space using a partially controlled movement subsystem. However, these types of uncertainty are not necessarily present only in isolation. In each specific problem of spatial self-organization, different combinations of these uncertainties can take place.

Evaluation of the efficiency of spatial self-organization algorithms usually is tied to the corresponding tasks of spatial self-organization. At the same time, emphasis is placed on minimizing the resources spent on solving the problem. Examples of resources include energy resources (battery charge), computing resources, communication resources, and others. In addition, in most cases, not one but a system of several performance criteria is used, among which those criteria that are of most interest to the user have the greatest weight. The following main criteria for evaluating the efficiency of solving spatial self-organization tasks can be distinguished: W_1 – time to complete the task; W_2 – consumption of energy resources for solving the problem; W_3 – consumption of computing resources; W_4 – the load on the information interaction system (consumption of communication resources); W_5 – information resource consumption; W_6 – an indicator of the completeness of the solution of the problem. In some cases, it is advantageous to compare the operation of the spatial self-organization algorithm with the corresponding ideal (reference) solution, which is usually modeled using a computational experiment.

IV. VARIANTS OF THE MOBILE AGENT ROBOTIC PLATFORM CONFIGURATION

The implementation of spatial self-organization algorithms depends on the type of robotic platform, in particular, on the method of organizing the movement of an agent in space, the type of environment in which it moves, the method of navigation, etc. [9-11]. All these attributes of the task affect the composition and type of data received as input by the spatial self-organization algorithm. These attributes determine the composition and type of data and commands that will come from the spatial self-organization algorithm to the mobile robotic platform.

Table 1

Types of uncertainty in the problems of spatial self-organization

	Type of uncertainty	Example
U(S)	Lack of information about the spatial characteristics of the environment	Unknown configuration of territory boundaries in problems of deploying a multi-agent system in space
U(A)	Lack of information about the actions of other agents (due to decentralized control)	Uncertainty about the further direction of movement of other agents in the problems of coordinated movement of a team of agents
U(P)	Lack of information about the dynamic characteristics of the environment that affect the movement of agents	Uncertainty about the direction and intensity of air flows in the problem of organizing the movement of paragliding robots

At the same time, it can be noted that the structure and parameters of spatial self-organization algorithms depend differently on different attributes of the task. That is, for each of the attributes, it is possible to determine a specific degree of dependence of the spatial self-organization algorithm on this attribute.

The problem of the dependence of the implementation of spatial self-organization algorithms on the main attributes of the corresponding tasks appears, for example, when these algorithms need to be transferred to a mobile robotic platform of another type. At the same time, from the point of view of the task itself, mobile agents perform the same tasks, while the conditions and methods for performing these tasks change. This situation can be illustrated by the example of the task of exploring a certain territory by a group of mobile agents to search for a target object. The algorithm for solving this problem in terms of organizing the coordinated collective behavior of agents will be almost the same for both ground-based wheeled robots and autonomous underwater vehicles jointly exploring a given section of the ocean floor. At the same time, the specific implementations of the algorithm in both cases will be very different due to the large difference between the corresponding mobile robotic platforms.

From the point of view of the influence of different variants of the mobile robotic platform configuration on the implementation of spatial self-organization algorithms, five main parameters can be distinguished that characterize the differences between these variants:

1) the ability of an agent to control its movement (movement subsystem);

2) navigation method (navigation subsystem);

3) capabilities of information interaction with other agents (communication subsystem);

4) the ability to perceive the environment (sensor subsystem);

5) decision-making capabilities (decision-making subsystem).

Let's denote each of these parameters as x_i, thus forming the set of parameters $X=\{x_1,x_2,x_3,x_4,x_5\}$. In this case, each parameter x_i can take values from the set $V=\{v_{ij}\}$, where *i* is the index of the parameter, and *j* is the index of the parameter value. The specific value v_{ij} of the parameter determines the composition and type of input data of the spatial self-organization algorithm and its output data in the form of commands to the mobile robotic platform. Taking this into account, a universal set of spatial self-organization algorithms should be independent of the set of parameter values V.

V. A SET OF SPATIAL SELF-ORGANIZATION ALGORITHMS INDEPENDENT OF THE ROBOTIC PLATFORM

An analysis of the main areas of work on the development of collective behavior algorithms that do not depend on the robotic platform allows to conclude that their main idea is to provide the algorithm with the same data, regardless of the type of robotic platform. That is, we are talking about different ways of adapting the composition and type of input and output data to the algorithm. As an alternative to this approach, we can propose the adaptation of the algorithm to the composition and type of data with which it works. That is, the algorithm must be able to work with different input data and provide the result of work in different forms. An interesting option within the framework of this approach may be the situation when the composition and type of input and output data are not known in advance, and as a result, the necessary adaptation process of the algorithm occurs directly in the process of its operation.

Let some spatial self-organization algorithm A be divided into separate functional blocks in such a way that the implementation of each block will depend on the value of one parameter of the robotic platform x_i (for example, only on the navigation method). Then algorithm A can be represented as a set of functional blocks a_{kj} executed sequentially: $A=\{a_{kj}\}$, where *k* is the index of the functional block, and *i* is the index of the parameter x_i, the value of which determines the implementation of this block. Thus, instead of algorithm A, the implementation of which depends on the whole set of parameters X of the robotic platform, we will get separate functional blocks of the algorithm a_{kj}, the implementation of each of which depends on the value only of one parameter x_i of the robotic platform.

In this case, for each value v_{ij} of the parameter x_i, one can implement a suitable variant of the functional block of the algorithm a_{kj}, which depends on x_i. Thus, we get a set of different implementations of the algorithm block a_{kj} for different values of v_{ij}: $r_v(a_{kj},v_{ij})$. If we implement all the blocks of the algorithm in this way, we will get a set of implementations of the blocks of the algorithm: $R_a=\{r_v(a_{kj},v_{ij})\}$, that is, the implementation of algorithm A for various robotic platforms. Within the framework of this approach, a specific robotic platform of an agent is specified as a set of parameter values on which the algorithm depends: $P=\{(x_i,v_{ij})\}$. The process of adapting the algorithm A to the given configuration of the robotic platform P_A is given by the functional transformation F: $(P, R_a) \rightarrow A^*$. The complexity and type of the functional transformation F are determined by the specifics of the task of adapting the spatial self-organization algorithm to the agent's robotic platform.

VI. THE PROBLEM OF COSCHEDULING SPATIAL SELF-ORGANIZATION AND DISTRIBUTED DATA COLLECTION

Spatial self-organization algorithms work with the input data they receive from the corresponding subsystems of the mobile robotics platform. Examples of input data are the location of the mobile agent, the location of other mobile agents of the team, certain data about the environment, etc. Examples of output data are coordinates of the point in space to which the mobile agent has to move, the direction of motion, motion speed, etc. It should be noted that the input/output data of spatial self-organization algorithms and the principles of their operation strongly depend on the type of mobile robotics platform. This dependence complicates the research and development of methods for organizing distributed data collection processes that control the behavior of a mobile agent in this aspect. Because of for each specific type of mobile robotics platform and the corresponding type of environment (land, air, water, etc.) it is necessary to develop and implement its specific version of the method for organizing distributed data collection processes. Therefore, there is a problem in finding and developing such methods of coscheduling spatial self-organization of mobile agents and distributed data collection, which would allow unification of the interaction of the mobile agents' motion control processes and those of distributed data collection, and reduce the functional dependence between them to a minimum. For this, it is necessary to develop such a method of coscheduling spatial self-organization and distributed data collection, which would provide the necessary level of functional coordination using a unified protocol of their interaction and an algorithm for planning their parallel execution.

VII. THE METHOD OF COSCHEDULING SPATIAL SELF-ORGANIZATION AND DISTRIBUTED DATA COLLECTION

To solve the problem, a method of coscheduling distributed data collection and spatial self-organization has been developed through coordinated parallel execution of the corresponding adaptive data collection process p(a) and the process of controlling the motion of a mobile agent p(s), which implements methods of spatial self-organization. The coordinated parallel execution of p(a) and p(s) is organized using the protocol $\Pi(a,s)$ of interaction between them and the scheduling algorithm $\Lambda(a,s)$. The proposed method of coscheduling implements functional decomposition in the following form (Fig.1):

$p(a,s) \rightarrow [p(a),p(s)]$,

$(a,s) \rightarrow C(a), C(s)$,

$W(a,s) \rightarrow W(a), W(s)$,

where

C(a) is a procedure for coordination at the level of joint data collection actions of mobile agents,

C(s) is a coordination procedure at the level of solving spatial self-organization tasks,

W(a) is a procedure for optimizing the parameters of the adaptive data collection process,

W(s) is a procedure for optimizing the parameters of the process of controlling the motion of a mobile agent.

The interaction of the adaptive data collection process p(a) and its subordinate motion control process p(s) within a mobile agent is carried out according to the protocol (Fig.1):

$\Pi(a,s)=\{D; Y(d), Y(s)\}$,

where

$D=\{d(x)\}$ are commands from the process p(a) to the process p(s) with parameters x, with which the adaptive data collection process p(a) implements the logic of joint data collection actions of mobile agents using their ability of spatial self-organization,

Y(d) are messages about the command execution status from p(s),

Y(s) are messages about the status of the motion control process p(s).

The set of functional relationships between p(a) and p(s) is reflected in the elements of the protocol $\Pi(a,s)$, in particular, in commands D and messages Y(d), Y(s).

The distribution of communication resources r(C) is given by the function $f_c(a,s)$. The distribution of computational resources r(p) between p(a) and p(s) is given by the function $f_p(a,s)$ and is provided by the scheduling algorithm $\Lambda(a,s)$:

$\Lambda(a,s)=\{(P_s(a),P_s(s),P_d,c), (T,m,L), (U_p,U_c,U_T)\}$,

where

$P_s(a)$ and $P_s(s)$ are the static priorities of the threads of processes p(a) and p(s), respectively, with

$P_s(a) < P_s(s)$;

P_d is the dynamic priority of a thread;

c is an estimate of the time during which the thread was executed in the processing unit;

T is a time slice that is allocated for the execution of the thread;

m is a time slice divider;

L is an indicator of the current load of the processing unit in the form of the average number of threads in its ready queue during time T;

U_p is a dynamic priority calculation rule that is applied to each thread in the ready queue every T second (Fig.2):

U_p: $P_d(a)=P_s(a)+c(a)$, $P_d(s)=P_s(s)+c(s)$;

U_c is a rule for calculating c(a) and c(s), which is applied every t=T/m second for the thread that is in the processing unit:

U_c: $c(a)=c(a)+2$, $c(s)=c(s)+1$;

U_T is a rule for calculating c for the threads in the ready queue, which is applied every T second:

U_T: $c=(c\times L)/(L+1)$.

In the proposed scheduling algorithm $\Lambda(a,s)$, every T seconds, the thread with the highest dynamic priority is selected for execution from the ready queue. At the same time, threads of the process p(s) have a higher static priority and are dispatched on a smaller timescale.

The proposed method of coscheduling p(a) and p(s) allows to increase the scale of parallelization of computations in the decision-making module of a mobile agent due to the increase in the maximum possible number of parallel threads H_p due to the functional decomposition $p(a,s)\rightarrow[p(a),p(s)]$.

VIII. SIMULATION RESULTS

In a situation where coscheduling with functional coordination is not used (that is, for p(a,s)), the maximum possible number of parallel threads can be estimated as $H_p=\max(k,n)$, where k is an estimate of the parallelization capabilities of computations for the adaptive data collection process p(a) (for example, in the form of the number of information sources k=M/N, which account for one data collection process), n is an estimate of the parallelization capabilities of computations for the movement control process p(s) (for example, in the form of the average number of neighboring mobile agents with whom this agent coordinates its movements). In contrast, in the case of coscheduling with functional coordination (that is, for [p(a), p(s)]), the maximum possible number of parallel threads can be estimated as $H_p=k\times n$. Based on this, it is also possible to estimate the share of consecutive instructions in one control loop for both cases:

1) for p(a,s):

$\alpha_1 = \alpha_k = (x_a + nx_s) / (kx_a + nx_s)$,

if k>n, and

$\alpha_1 = \alpha_n = (x_s + kx_a) / (kx_a + nx_s)$,

if k<n, where x_a, x_s are the numbers of consecutive instructions of the control loop in one thread, respectively, of processes p(a) and p(s);

2) for [p(a), p(s)]:

$\alpha_2 = (x_a + x_s) / (kx_a + nx_s)$.

According to Amdahl's law, since $\alpha_2<\alpha_k$ and $\alpha_2<\alpha_n$, the proposed method of coscheduling gives a gain in speeding up computations in the decision-making module of the mobile agent due to more efficient parallelization. This gain can be estimated using the relative acceleration of computations $g=S_p(\alpha_1,p)/S_p(\alpha_2,p)$, where $S_p(\alpha_1,p)$ is the acceleration of computations for the case p(a,s), and $S_p(\alpha_2,p)$ is the acceleration of computations for the case [p(a), p(s)].

Simulation results of the proposed method of coscheduling show that the relative acceleration of computations for a given number *p* of processing units and values ($x_a > x_s$) increases nonlinearly with increasing (k, n) to the maximum value in case of $k\approx n$, then decreases (Table 2). Therefore, for each combination of values (n, k, x_a, x_s) there is a corresponding number of processing units *p*, for which the proposed method gives the maximum relative acceleration of computations. On average, for typical values (n, k, p, x_a, x_s), the proposed method provides a 40.6 % acceleration of computations in the decision-making module of the mobile agent (arithmetic average of experimentally obtained estimates of relative acceleration of computations for all combinations of values (n = {2,…,5}, k={2,…,10}, *p*={2, 4, 6, 8}, x_a ={6, 8, 10, 12}, x_s ={6, 8, 10, 12}).

IX.CONCLUSION

The problem of coscheduling spatial self-organization control processes and distributed data collection processes in a multi-agent system was considered. The goal of coscheduling was to find and use the possibilities of functional coordination of these processes and increase the efficiency of the multi-agent system due to their parallel execution.

An analysis of the main features of spatial self-organization tasks that affect the solution of the problem of coscheduling was conducted. Three main types of uncertainty, which complicate the solution of spatial self-organization problems: lack of information about the spatial characteristics of the environment, lack of information about the actions of other agents, and lack of information about the dynamic characteristics of the environment that affect the movement of agents, were identified. Methods for evaluating the efficiency of spatial self-organization algorithms were given. Variants of the mobile agent robotic platform configuration and the problem of the dependence of spatial self-organization algorithms on the type of robotic platform were considered. An approach to the development of a set of algorithms for spatial self-organization independent of the robotic platform of a mobile agent was proposed due to the functional decomposition of these algorithms.

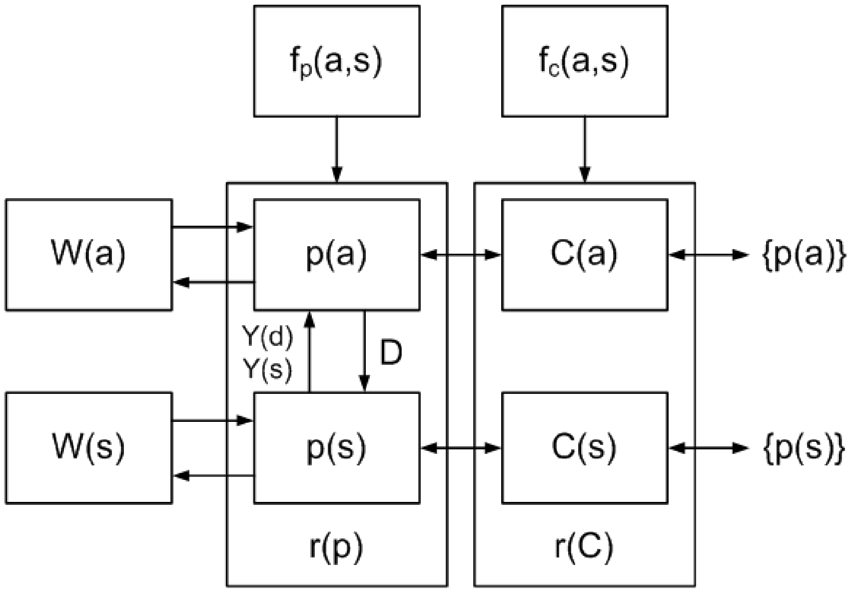

Fig. 1. The diagram of coscheduling spatial self-organization of mobile agents and adaptive processes of distributed data collection.

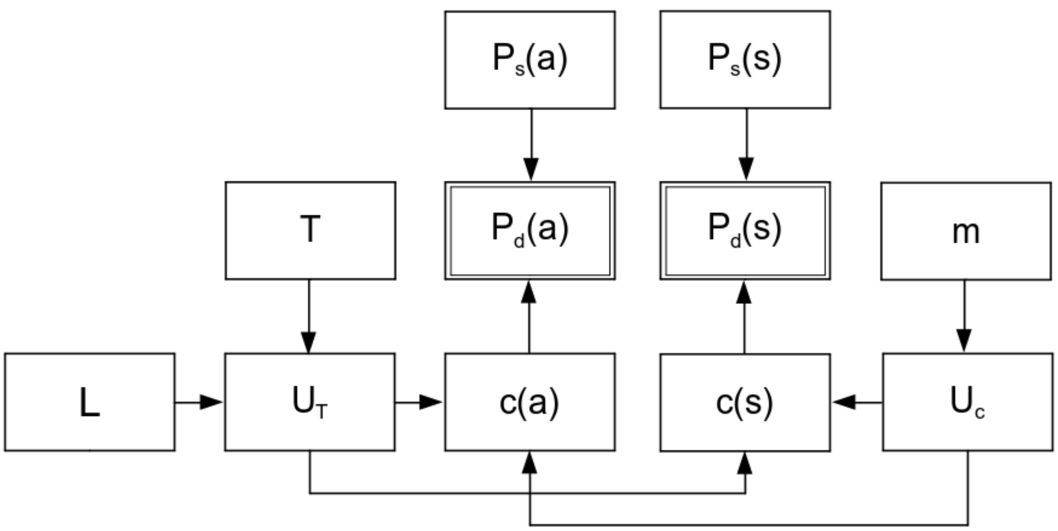

Fig. 2. The diagram for calculating dynamic priorities $P_d(a)$ and $P_d(s)$.

Table 2

The relative acceleration of the computations $g=S_p(\alpha_1,p)/S_p(\alpha_2,p)$, where $\alpha=\alpha_1$ or $\alpha=\alpha_2$, $p=4$, $x_a=10$, $x_s=8$ (g – calculation, g^* – estimate obtained experimentally)

	k=2		k=3		k=4		k=5		k=6		k=7		k=8		k=9		k=10	
	g	g^*	g	g^*	g	g^*	g	g^*	g	g^*	g	g^*	g	g^*	g	g^*	g	g^*
n=2	1.27	1.26	1.24	1.22	1.22	1.21	1.20	1.18	1.18	1.17	1.17	1.15	1.16	1.14	1.15	1.13	1.14	1.13
n=3	1.31	1.30	1.44	1.43	1.40	1.39	1.38	1.37	1.35	1.33	1.32	1.31	1.30	1.29	1.29	1.28	1.27	1.26
n=4	1.28	1.26	1.52	1.50	1.57	1.55	1.53	1.52	1.49	1.47	1.46	1.44	1.43	1.41	1.41	1.40	1.39	1.37
n=5	1.26	1.24	1.48	1.45	1.67	1.66	1.67	1.65	1.62	1.61	1.59	1.58	1.55	1.53	1.52	1.51	1.49	1.47

The problem of coscheduling spatial self-organization and distributed data collection was analyzed. A method of coscheduling of spatial self-organization and distributed data collection by coordinated parallel execution of the corresponding data collection process and the process of controlling mobile agent motion was proposed. The method of coscheduling was implemented using the interaction protocol of these processes and the algorithm for planning their parallel execution using functional decomposition. The simulation results of the proposed method of coscheduling were given. It was proved that the proposed method of coscheduling provides acceleration of computations in the decision-making module of the mobile agent due to more efficient parallelization. On average, for typical values of parameters of control processes, the proposed method of coscheduling provides an acceleration of computations in the decision-making module of the mobile agent by 40.6 %.

References

[1] Dorri, A., Kanhere, S., Jurdak, R. (2018). Multi-Agent Systems: A Survey, in IEEE Access, vol. 6. – pp. 28573-28593, DOI: 10.1109/ACCESS.2018.2831228.

[2] Rizk, Y., Awad, M., Tunstel, E. (2018). Decision Making in Multi-Agent Systems: A Survey, in IEEE Transactions on Cognitive and Developmental Systems, vol. 10, no. 3. – pp. 514-529, DOI: 10.1109/TCDS.2018.2840971.

[3] Arup Kumar Sadhu, Amit Konar (2020). Multi-Agent Coordination: A Reinforcement Learning Approach, Wiley. – 320 p.

[4] Botchkaryov, A., Golembo, V., Paramud, Y., Yatsyuk, V. (2019). Cyber-physical systems: data collection technologies, A. Melnyk (ed.), Lviv, «Magnolia 2006». – 176 p. (in Ukrainian).

[5] Shermin, A., Dhongdi, S. (2022). Review of Underwater Mobile Sensor Network for ocean phenomena monitoring, Journal of Network and Computer Applications, Volume 205, Issue C. – pp. 1-24. DOI: 10.1016/j.jnca.2022.103418.

[6] Wang, Z., Li, H.X., Chen, C. (2020). Reinforcement Learning-Based Optimal Sensor Placement for Spatiotemporal Modeling. IEEE Transactions on Cybernetics; 50(6). – pp. 2861-2871. DOI: 10.1109/TCYB.2019.2901897.

[7] Song, M., Hu, C., Gong, W., Yan, X. (2022). Domain Knowledge-Based Evolutionary Reinforcement Learning for Sensor Placement. Sensors (Basel); 22(10):3799. – pp. 1-17. DOI: 10.3390/s22103799.

[8] Meng, X., Inaltekin, H., Krongold, B. (2019). Deep Reinforcement Learning-Based Topology Optimization for Self-Organized Wireless Sensor Networks, 2019 IEEE Global Communications Conference (GLOBECOM). – pp. 1-6, DOI: 10.1109/GLOBECOM38437.2019.9014179.

[9] Krishnan, V., Martínez, S. (2018). Distributed Control for Spatial Self-Organization of Multi-agent Swarms, SIAM Journal on Control and Optimization, vol. 56, Iss. 5. – pp. 3642-3667. DOI: 10.1137/16M1080926.

[10] Liu, Y., Bucknall, R. (2018). A survey of formation control and motion planning of multiple unmanned vehicles. Robotica. 36. – pp. 1-29. DOI: 10.1017/S0263574718000218.

[11] Sun, Z. (2018). Cooperative Coordination and Formation Control for Multi-agent Systems. Springer Cham. – 179 p. DOI: 10.1007/978-3-319-74265-6.

[12] Krishnan, V., Martínez, S. (2022). A multiscale analysis of multi-agent coverage control algorithms, Automatica, Volume 145,110516.- pp. 1-26. DOI: 10.1016/j.automatica.2022.110516.

[13] Hu, J., Bhowmick, P., Jang, I., Arvin F., Lanzon, A. (2021). A Decentralized Cluster Formation Containment Framework for Multirobot Systems, in IEEE Transactions on Robotics, vol. 37, no. 6. – pp. 1936-1955, DOI: 10.1109/TRO.2021.3071615.

[14] Hu, J., Bhowmick, P., Lanzon, A. (2021). Group Coordinated Control of Networked Mobile Robots with Applications to Object Transportation, in IEEE Transactions on Vehicular Technology, vol. 70, no. 8. – pp. 8269-8274, DOI: 10.1109/TVT.2021.3093157.

[15] Hu, J., Turgut, A., Lennox B., Arvin, F. (2022). Robust Formation Coordination of Robot Swarms with Nonlinear Dynamics and Unknown Disturbances: Design and Experiments, in IEEE Transactions on Circuits and Systems II: Express Briefs, vol.69, no.1. – pp. 114-118, DOI: 10.1109/TCSII.2021.3074705.

[16] Amrani, N. E. A., Snineh, S. M., Youssfi, M., Abra, O. E. K., Bouattane, O. (2021). Interoperability model between heterogeneous MAS platforms based on mobile agent and reinforcement learning, 2021 Fifth International Conference on Intelligent Computing in Data Sciences (ICDS). – pp. 1-8, DOI: 10.1109/ICDS53782.2021.9626723.

Alexey Botchkaryov was born in 1975 in Lviv, Ukraine. He received B.S. and M.S. degrees in Computer Engineering at Lviv Polytechnic National University, in 1998 and a 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. Currently, he is an associate professor at the Computer Engineering Department, ~~at~~ Lviv Polytechnic National University. His research interests include self-organization in complex systems, structural adaptation, intelligent information-gathering agents, and multi-agent systems.