

Algorithms for changing the structure of geospace self-organizing question-answering sensor networks

Vladimir Mochalov^{1*}, Anastasia Mochalova¹

¹Institute of Cosmophysical Research and Radio Wave Propagation FEB RAS, 684034 Kamchatka region, Elizovskiy district, Paratunka, Russia

Abstract. Optimization problems of construction, development and changing the structure of geospace self-organizing question-answering sensor networks (GSQASN) are considered. The task specifies the coordinates of various network nodes. It is required with the specified functional, structural, cost and spatial constraints to change the structure of the GSQASN by adding new nodes, moving to new positions or removing some existing nodes. After the formation of the GSQASN structure we solve the task of question-answer agents placement into GSQASN structure in order to be able to answer the given types of questions under the established limitations. The functional scheme of a given category nodes placement into GSQASN structure and approximate bio-inspired algorithms for solving the tasks are proposed. The results of the work can be used in the construction of specific GSQASN and in the GSQASN design support systems.

1 Introduction

New space technologies (nanosatellites, CubeSats, SmallSats and etc.), private space companies and the projects for launching thousands of small satellites to organize space networks with different purposes give principally new opportunities to monitor geospheres. We should note the increasing number of separate monitoring systems applying the data obtained from the geospace. The examples of such systems are the following: systems of estimation of magnetospheric parameters based on whistler distributed registration (AWDANet system); distributed systems for lightning activity monitoring (WWLLN, Blitzortung.org, Earth Networks and etc.); the Earth's ionosphere monitoring systems (ionospheric vertical sounding systems; GPS/GNSS ionospheric sounding allows to get the data about variations of the total electron content (TEC) in the ionosphere with high spatial and temporal resolution; and etc.), different monitoring systems applying satellite data (including global navigational systems to determine locations and time synchronization of measurements), space weather monitoring systems and so on. The term «geospace» is understood as the region of space that goes from the solar photosphere to the atmosphere of

* Corresponding author: vmochalov@ikir.ru

Earth. It includes the solar photosphere, chromosphere and corona, the solar wind, Earth's magnetosheath, magnetosphere, thermosphere, ionosphere and atmosphere.

In this paper on the basis of the personal results obtained before in the area of sensor networks designing [1-4], semantic analysis and question-answering systems [5-7] we introduce a new notion of «geospace self-organizing question-answering sensor networks» (GSQASN) that means a distributed network which monitors ambient environment parameters applying the data from geospace and allowing nodes not only to interact with each other owing to the self-organizing property, but to answer the defined types of natural language questions as well.

In the work we use a model of the GSQASN structure (Fig.1), where on the functional level the following types of GSQASN nodes can be defined: (1) functional nodes (F-nodes) that collect information in some neighborhood of their location; (2) transit nodes (T-nodes) that manage routing and retransmit the information collected by F-nodes to the information collection centers (ICC) to be utilized further; (3) ICCs that manage the GSQASN and process information collected by the GSQASN. In general case there can be multiple ICCs in the GSQASN, and the information that has arrived into each of them is available to one or multiple users for making decisions and performing certain actions. It means that information received by F-nodes should be retransmitted, with a required degree of reliability, to several ICCs by means of transit nodes allocated within the given object in a certain way. We think that ICC is capable of performing F-node and T-node functions. F-node can perform the T-node functions and information between nodes can be transmitted both via the wire and wireless networks.

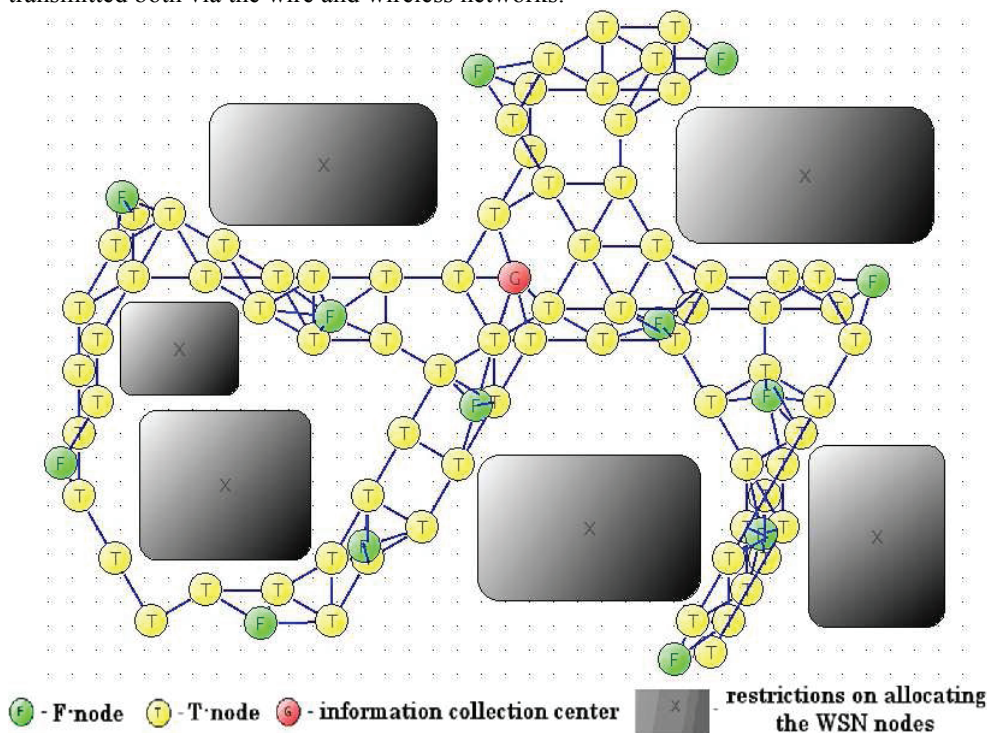


Fig. 1. Example of a distributed fault-tolerant GSQASN structure.

2 The functional scheme of a given category nodes placement into GSQASN structure

Designing of GSQASN requires the solution of many complicated problems referring to different areas of research, they are: projecting of network nodes (measurement stations, sensors and etc.); construction of different physical-mathematical models of monitoring processes of ambient environment parameters applying the geospace data; model selection for information collection from the GSQASN; development of methods and algorithms for GSQASN structure synthesis; estimate of measurement error and limitations, estimate of spatial and other limitations of network node placement; ensuring of the defined functional and structural parameters of the synthesized GSQASN; development of fitness functions of network nodes placement; development of algorithms for question-answering agents placement into the GSQASN structure; development of self-organization algorithms for different GSQASN levels (for example, the function level performed by the nodes, routing level, level of tasks distribution between nodes and question-answering agents).

Two types of GSQASN structures can be distinguished:

- distributed fixed network in which all the GSQASN nodes are not moved after the initial placement;
- decentralized mobile self-organizing network compound of fixed nodes (distributed fixed network segment) and mobile units (mobile network segment) which can be moved in different directions and, as a consequence, form different network structures in dynamics by break and establish network connections with other nodes, removal and installation of new nodes into the network structure. Mobile units can be installed both on satellites and on mobile robots (drones, above-water and under-water vehicles). Some functions of the mobile units are: formation of a self-organizing GSQASN structure; geographically-distributed acquisition of data from GSQASN nodes; organization of the interaction of a GSQASN mobile network segment with a fixed one. When applying the mobile robots, the following is possible: accurate nodes placement, distribution nodes over territory; moving, removing, reprogramming of nodes; charging and replacement of GSQASN nodes power sources; planning of cooperative behavior of mobile robots in the process of general aim solution and redesigning of GSQASN nodes placement based on the aims and current measurements of the whole GSQASN.

Types of node structures of a distributed fixed GSQASN are: star (Fig.2); cluster network structure (Fig.3); mesh self-organizing structure (Fig.1).

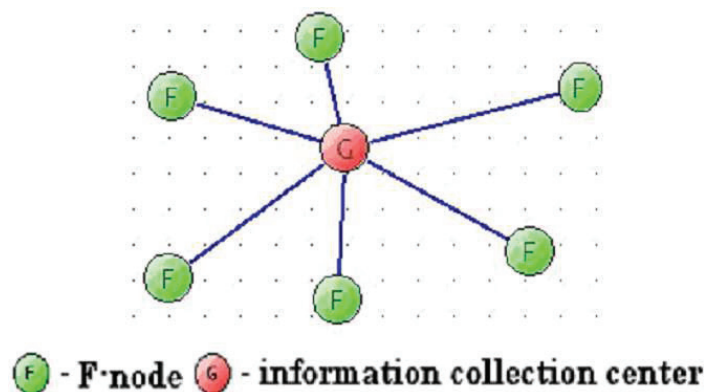


Fig. 2. Star network structure.

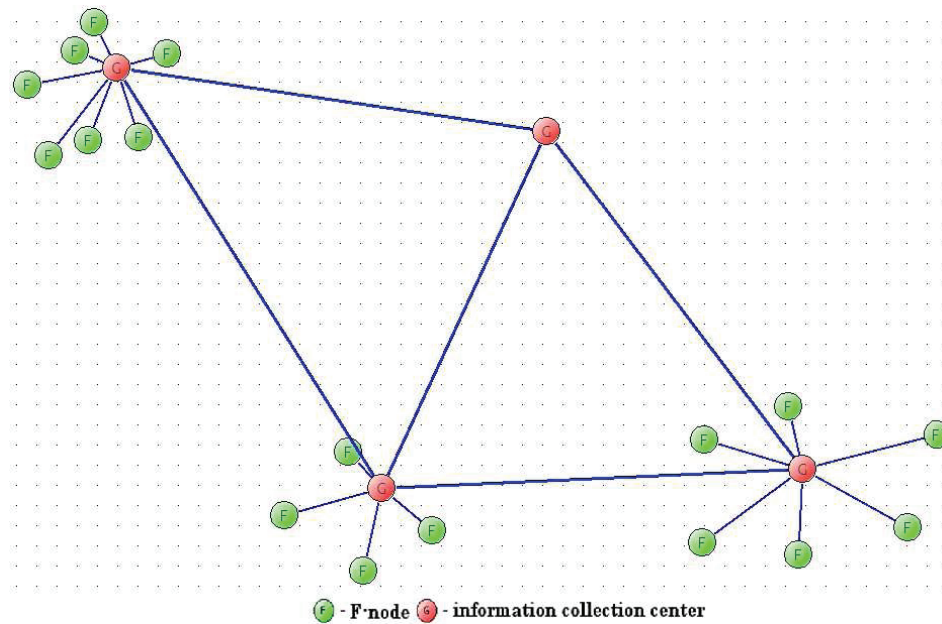


Fig. 3. Cluster network structure.

Possible ways of problem statement for GSQASN structure synthesis

1. Synthesis of ICCs allocation. In this task we know the spatial restrictions for allocating the ICCs. Also, we know the allocation of pre-installed ICCs. It is necessary to allocate ICCs in such way, that the designed GSQASN structure would have the «desired properties» assigned by a designer. During the synthesis of ICCs allocation it is possible to optimize (by their removal or moving) some pre-installed ICCs noted by the designer.
2. Synthesis of F-nodes allocation. We know the spatial restrictions for allocating the F-nodes. Also, we know the allocation of ICCs and allocation of pre installed F-nodes. It is necessary to allocate new F-nodes in such way, that the designed GSQASN structure would have the «desired properties» assigned by a designer. During the synthesis of F-nodes allocation it is possible to optimize (by their removal or moving) some pre-installed F-nodes noted by the designer.
3. Synthesis of T-nodes allocation. We know the description of the GSQASN allocation object, spatial restrictions for allocating the T-nodes. Also, we know the allocation of F-nodes, ICCs and pre installed T-nodes. It is necessary to allocate T-nodes in such way, that the designed GSQASN structure would have the «desired properties» assigned by a designer. During the synthesis of T-nodes allocation it is possible to optimize (by their removal or moving) some pre-installed T-nodes noted by the designer.
4. Complex sequential synthesis of ICCs, F-nodes and T-nodes allocation. This statement suggests a sequential allocation of ICCs first (statement 1), then F-nodes (statement 2) and then T-nodes (statement 3).

Fig. 4 illustrates a functional scheme of a given category nodes placement into GSQASN structure. This scheme can be used as the basis for synthesis of ICC, F-nodes and T-nodes placement.

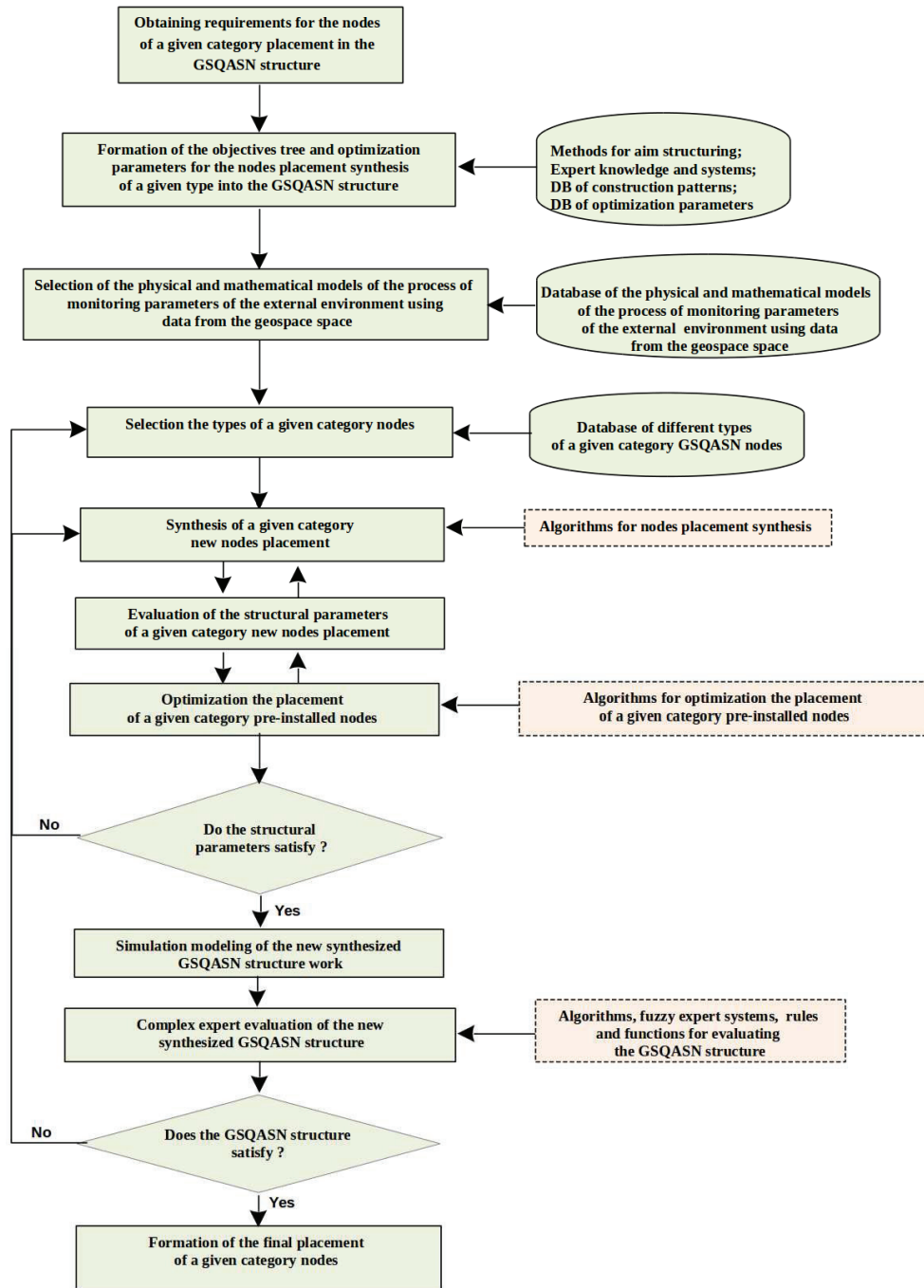


Fig. 4. The functional scheme of a given category nodes placement into GSQASN structure.

3 Bio-inspired algorithm for synthesis of a given category nodes placement into GSQASN structure

In recent years, the research area of *Natural Computing* is rapidly developing. It unites mathematical methods in which the principles of natural mechanisms of decision making are embedded [8]. Scientists have developed bio-inspired algorithms (BA) of modeling animals behavior ([9], [10], etc.) for solving various optimization problems that either do not have exact solution, or the solutions search space is vary large and complex constraints of the objective function are presented, as well as NP-complete.

The described recommendations on applying BA and the proof in [11] that even the constrained variant of the problem of minimal coverage on plane is NP-complete allow us to conclude about the possibility to apply self-organizing bio-inspired algorithms for a self-organizing GSASN structure design.

Bio-inspired algorithms can be seen as multi-agent systems, each agent in which operates autonomously on very simple rules. The most frequently used bio-inspired agents (B-agents) include: ants, bees, termites, fireflies, birds, fish, bats, cats, wolves, etc.

The initial data of the algorithm for synthesis of a given category nodes $type_x$ placement are the following: allocation of pre-installed nodes of $type_x$ (set of nodes Ω_{pin}); allocation of installed nodes of $type \neq type_x$; description of the object that the GSQASN needs to be located at (its dimensions, scheme, spatial requirements for nodes allocation and etc.); characteristics of ready-to-use nodes of $type_x$; adopted self-organization and routing algorithms; information collection model; functional requirements; optimization parameters; fuzzy expert systems etc.

The algorithm below is based on the adaptation of the multi-agent bio-inspired algorithm for wireless sensor network design proposed in paper [1].

Step 1: Define a set M_{ALL} of all optimization parameters; the functions for calculating the parameters of M_{ALL} ; a subset of optimization parameters $M_i \in M_{ALL}$,

Step 2: Determine the membership functions of fuzzy sets that characterize the optimization parameters of the M_{ALL} ; fuzzy expert system to derive the confidence factor to meet the functional requirements of the designer. Create an empty set of the best solutions Ω_{BEST} . Determine the maximum number of solutions b_K that will be stored in Ω_{BEST} ,

Step 3: Create a set Ω_p of the possible placement points of type $type_x$ nodes (the set Ω_p can be formed with the help of: the algorithms for covering the object of placement with a mesh (based on an equilateral triangle or hexagon or square) or with circles with a given radius; covering algorithms in accordance with the choice and recommendations of the designer; other covering algorithms). Create an empty set Ω_T . Create node of type $type_x$ in each point of Ω_p and add this node to Ω_T set,

Step 4: Create a shared global memory of the stored pheromone (SGMSP) to share some "knowledge" between B-agents. The pheromone is stored on the edges of a fully connected undirected weighted graph (FCUWG), the nodes of which are the ones of $type_x$. To store the edges of the graph in computer memory it is required to create a two-dimensional array *feromoneNetwork* with $N(N-1)/2$ memory cells of type *float*, where $N=|\Omega_T|$ is the number of nodes. All values of *feromoneNetwork* must be initialized as zeros,

Step 5: Execute bio-inspired multi-agent algorithms.

5.1. Create a two-dimensional array *feromoneDif* to store changes in the pheromone using for example the following Java-code:

```
float feromoneDif [][] = new float[N-1][];  
for(int i = 0; i < N; i++)  
feromoneDif [i] = new float[N - (i + 1)];
```

All values of *feromoneDif* must be initialized as zeros (in the above code, the zero-initialization is done automatically). Define the number of different bio-inspired agents *m*, the strategy for choosing the initial location of the agent and other parameters needed for the agent to perform the work.

5.2. For each agent, perform the following steps (the code can be parallelized, i.e. to run in a separate thread for each agent):

5.2.1. Form, using the movement rules of the B-agent, SGMSP the route M_T (array) of agent moving on nodes Ω_T ,

5.2.2. Create an empty extensible array of nodes M_{STR} , in which the nodes of the designed structure will be placed,

5.2.3. Select the design strategy:

(a) sequentially add nodes to the network -> Go to step 5.2.4;

(b) sequentially remove nodes from the network -> Add to M_{STR} all nodes from M_T array in the same sequence order. Go to step 5.2.11;

5.2.4. Create an empty set H_P , which will contain the caches of such internal parameters of the functions of computing estimates M_I , which will increase the speed of computing estimates M_I for the next iteration. Create a variable *i* to store the index of the current node from the M_T array and initialize its value to 0 ($i=0$). Set the node T_C ($T_C=M_T[0]$) as the current one,

5.2.5. Add to M_{STR} the node T_C . Form the network structure S_S of nodes M_{STR} ,

5.2.6. Calculate using the caches H_P the values estimations of the optimization parameters of the set M_I having structure S_S . Clear H_P . Save the caches of the internal parameters of the functions of computing estimates M_I of the current iteration to the set H_P ,

5.2.7. Using a fuzzy expert system, calculate the reliability coefficient K_{DI} of meeting the requirements of the designer for parameters of a set M_I of structure S_S ,

5.2.8. If $K_{DI} > p_I$, where p_I is a set threshold, go to step 5.2.11,

5.2.9. If $i < |M_T|$, put $i = i + 1$ and accept the next node $T_C = M_T[i]$ as the current one. Repeat steps 5.2.5. – 5.2.9. while *i* does not become equal to $|M_T|$,

5.2.10. Exit with notification of the failure from the function of agent design of the network structure,

5.2.11. Steps of eliminating optimization:

5.2.11.1. Select the strategy of eliminating optimization:

(a) step-by-step optimization with consideration of optimization parameters M_I . The fuzzy expert estimation of the structural parameters M_I is used;

(b) step-by-step optimization with consideration of all optimization options M_{ALL} . The unit of simulation modeling and complex assessment of the network is used;

5.2.11.2. Revert the M_{STR} array,

5.2.11.3. In a loop, temporarily exclude each node $T_O \in M_{STR}$ from M_{STR} , then compute the confidence factor K_D of meeting the requirements of the eliminating optimization strategy parameters. If in absence of node T_O evaluation of network structure stops meeting the designer requirements, put T_O back to M_{STR} into its place.

5.2.11.4. Revert the M_{STR} array,

5.2.12. Perform simulation modeling of the network. The results of the modeling and structural-parametric estimates of the various parameters are the input to the complex expert system for evaluation of network structure. Calculate with the latter the confidence factor K_{DALL} of meeting all the designer requirements,

5.2.13. If $K_{DALL} > 0$ then, in accordance with one of the following strategies, increase the pheromone amount in the array *feromoneDif*:

(a) consequent update – increase the amount of pheromone on the edges of the agent sequential traveling on nodes of M_{STR} by the value equal to $\Delta \tau_{ij,k}(t) = Q_{agent} * K_{DALL}$, where Q_{agent} is the amount of pheromone secreted by the agent on one edge;

(b) **full-mesh update** – increase the amount of pheromone on all edges of the fully connected graph constructed on the basis of nodes of M_{STR} by the value equal to $\Delta \tau_{ij,k}(t) = Q_{agent} * K_{DALL}$.

5.2.14. If K_{DALL} is greater than the estimate of the worst solution from Ω_{BEST} , or ($|\Omega_{BEST}| < b_K$ and $K_{DALL} > 0$), then add into Ω_{BEST} the current solution. By the solution we mean the couple (M_{STR}, K_{DALL}) . If $|\Omega_{BEST}| \geq b_K$ then leave in Ω_{BEST} only b_K best solutions.

5.3. After all agents have performed step **5.2**, update SGMSP (*feromoneNetwork* array) in accordance with the following well-known rule [9]: $\Delta \tau_{ij}(t+1) = (1-p) * \tau_{ij}(t) + \Delta \tau_{ij}(t)$, where $\Delta \tau_{ij}(t)$ is the amount of pheromone on edge (i,j) in the array of pheromone changes *feromoneDif*, and $p \in [0,1]$ is the coefficient of pheromone evaporation. To enhance the intermediate best solutions, the amount of pheromone on the edges of the routes of the best solutions Ω_{BEST} should be increased (an example is using "elite" ants).

5.4. If the stopping criterion is not met, go to step **5.1**.

Step 6: If it is necessary to continue the search, then create new set Ω_T and add to it type $type_x$ nodes located in points of possible nodes placement (for example, to cover the object of nodes placement with a mesh more densely in comparison with the previous coverage). After it go to Step 4.

Step 7: Return the best solution from Ω_{BEST} .

4 Modifications of bio-inspired algorithm for changing the GSQASN structure

Under the change of the GSQASN structure we assume adding new nodes, moving to new positions or removing some existing pre-installed nodes with the specified functional, structural, cost and spatial constraints. Moving the node to new position is supposed to remove it from the old position and add it to a new one. The algorithm for adding new nodes to the GSQASN structure was considered in Section 3. Consider the algorithms of moving to new positions or removing some existing pre-installed nodes based on the modification of the algorithm discussed in Section 3.

Algorithm 1 for changing the GSQASN structure based on the modification of the algorithm discussed in Section 3

- At the beginning of the Step 4 add to the set Ω_T those pre-installed nodes of $type_x$ of set Ω_{pin} that can be removed or moved to new positions.
- In the implementation functions of GSQASN structure evaluation take into account the cost of removing and moving some pre installed nodes, as well as changing the cost of maintenance these nodes.
- All other steps of the bio-inspired algorithm for synthesis of a given category nodes placement into GSQASN structure remain unchanged.

Algorithm 2 for changing the GSQASN structure based on the modification of the algorithm discussed in Section 3

- Perform all steps of the algorithm considered in Section 3 so that new nodes of $type_x$ will be added to the GSQASN structure, but optimization of some pre-installed nodes placement of $type_x$ will not be performed.
- To optimize the placement of some pre-installed nodes of $type_x$ perform steps 1-5 of the algorithm discussed in Section 3 with the following modifications:

- a) On the step 3, add only those pre-installed nodes of $type_x$ from the set Ω_{pin} to the set Ω_T , which can be removed or moved to new positions (don't add other nodes to the set Ω_T).
- b) In the implementation of functions of the GSQASN structure evaluation take into account the cost of removing and moving of some pre-installed nodes, as well as changing of the cost of the node maintenance.

5 The scheme for generating the answer using the GSQASN question-answer agents

The question-answer agent (QA-agent) performs the function of generating the answer from natural language questions by collection, aggregation and accumulation information from some F-nodes that are serviced by this QA-agent. After the accumulation of sufficient information from the group of F-nodes, the QA-agent generates an answer. The QA-agents can interact with each other through self-organization to be able to answer the given types of questions under the established limitations. Physically, the QA-agent is a software/hardware add-on that can upgrade any type of GSQASN nodes. The QA-agents can differ from each other by technical capabilities (due to various hardware and software add-ons) and functionality capabilities (the ability to answer different types of questions, performed functions, etc.).

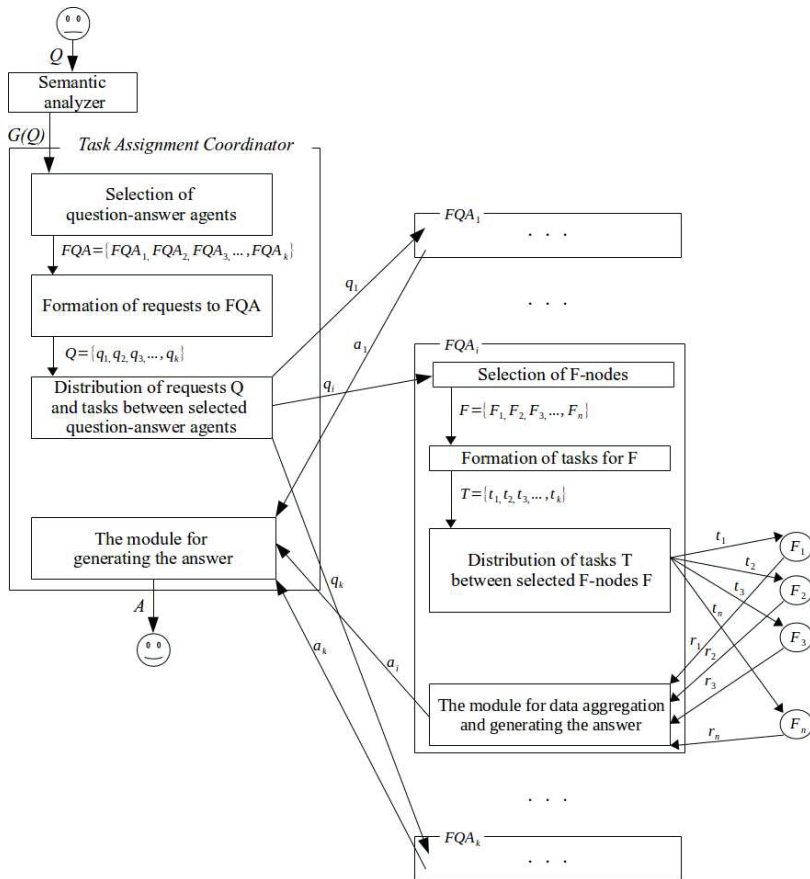


Fig. 5. The scheme for generating the answer using the GSQASN question-answer agents.

In general, formulated in the natural language question Q enters to the input of one of the QA-agents that perform the functions of task coordinator for other QA-agents. After it the question Q enters to the input of the semantic analyzer module, which create ontological-semantic graph $G(Q)$. The graph $G(Q)$ enters to the input of the module for selection of QA-agents, which are best suited for generating the answer to question Q . The information about selected QA-agents is placed into a set $FQA = \{FQA_1, FQA_2, FQA_3, \dots, FQA_k\}$. The set FQA enters to the input of generating requests for QA-agents module. As a result, for each QA-agent $FQA_i \in FQA$ this module form a request q_i . Each q_i request is transmitted to the QA-agent FQA_i , which first tries to find the necessary information in the local database and if it is not found FQA_i select serviced F-nodes from which the necessary information should be collected. The information of selected at this stage F-nodes is placed into the set $F = \{F_1, F_2, F_3, \dots, F_n\}$. Further, the task formation module generates a task t_j for each F-node $F_j \in F$. Each F-node $F_j \in F$ that receives the task t_j executes it (using information stored on this F-node or receive it from the environment with the help of a sensor installed on the F-node) and sends the response r_j back to the QA-agent which generated the task t_j . On the basis of all the responses $r_1, r_2, r_3, \dots, r_n$ obtained from F-nodes $F_1, F_2, F_3, \dots, F_n$ QA-agent FQA_i generates a_i answer and send it back to the task assignment coordinator. Thus, the task assignment coordinator receives all answers from QA-agents and on the basis of them makes up a general answer A , which is transmitted to the user as an answer to the question Q (Fig.5).

6 Question-answer agents placement into GSQASN structure

The selection of GSQASN nodes for assigning them additional QA-agents functions can be executed both at the stage of the GSQASN nodes placement synthesis (by adding the logic of QA-agents placement to the implementation functions of GSQASN structure evaluation) and after the GSQASN nodes have already been placed. In general, the task of QA-agents placement into the GSQASN structure is formulated as follows: with the well-known allocation of all GSQASN nodes (ICCs, F-nodes and T-nodes) we need to solve the optimization problem of selection some GSQASN nodes to assign them additional QA-agents functions to be able to answer the given types of questions under the established limitations.

Consider the algorithm for QA-agents placement into the GSQASN structure based on the modification of the algorithm discussed in Section 3. In this modification we synthesize the placement of QA-agents nodes (a given category nodes $type_x=QA$ -agent).

Algorithm for QA-agents placement into the GSQASN based on the modification of the algorithm discussed in Section 3

- On the step 3 create a set Ω_p of all possible GSQASN nodes for assigning them additional QA-agents. Create an empty set Ω_T . Create node of $type_x=QA$ -agent on each node of Ω_p and add this node to Ω_T set.
- To implement the functions of the GSQASN structure evaluation add logic for QA-agents placement evaluation.
- All other steps of the bio-inspired algorithm for synthesis of a given category nodes placement into GSQASN structure remain unchanged.

References

1. V.A. Mochalov, *Proceedings, The IEEE 17th ICACT*, 34-42 (2015)
2. V.A. Mochalov, *Lecture Notes in Computer Science* **9247**, 11-22 (2015)
3. V.A. Mochalov, A.P. Pschenichnikov, *ICACT TACT* **4**, 659-663 (2015)
4. V.A. Mochalov. Certificate of registration of computer software “Program for synthesis of monitoring networks by bio-inspired algorithms” No. 2016612039 (2016).
5. Mochalova A.V., Mochalov V.A, *Lecture Notes in Artificial Intelligence, Advances in Computational Intelligence*, **10061**, 53-66 (2017)
6. A.V. Mochalova, V.P. Zacharov, V.A. Mochalov, *ICACT TACT* **5**, 902-906 (2016)
7. V.A. Kuznetsov, V.A. Mochalov, A.V. Mochalova, *ICACT TACT* **5**, 651-658, (2016)
8. G. Rozenberg, T. Bäck, J.N. Kok, *Handbook of Natural Computing*, (2012).
9. F. Bonavear, M. Dorigo, G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Systems*, 320 p, (1999)
10. D. Teodorovic, *Bee Colony Optimization (BCO)*, Available:
<http://www.sf.bg.ac.rs/downloads/katedre/oi/1.BCO-Book-Chapter.pdf> .
11. R.J. Fowler, *Inf. Process. Lett.*, **12(3)**, 133-137 (1981).