Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System

Gennady Krivoulya[1], Volodymyr Tokariev[1], Iryna Ilina[1], Oleh Lebediev[1] and Vladislav Shcherbak[1]

[1] *Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine*

Abstract

The paper explores the possibility of using a centralized particle swarm algorithm to distribute subtasks between «s-bot» one «Swarm-bot» system to solve the main problem. Based on the centralized particle swarm algorithm, a sequence of iterations was developed, which showed that it is an effective algorithm, since it allows you to find the best solution to the problem much faster. It was found that the developed iteration algorithm based on the centralized particle swarm algorithm is distinguished by its simplicity of operation, a small set of input parameters that must be set at the first iteration, sufficiently acceptable accuracy and, which is especially encouraging, the developed algorithm has a fast convergence to making the optimal decision.

Keywords 1

Swarm-bot systems, s-bot, intelligent mobile «s-bot», intelligent embedded systems, internet of things, iteration algorithm, particle swarm algorithm, unorganized physical environment

1. Introduction

Currently, there is a tendency to complicate the existing and the emergence of fundamentally new «Swarm-bot» systems. This is due to the emergence of new architectures of «Swarm-bot» systems for various purposes, a variety of information flows, the design of fundamentally new intelligent embedded control systems in such areas as the defense industry, green energy, robotics, biomedical engineering, etc. The complication of new «Swarm-bot» systems is caused by the need to consider factors that interact with the environment, an increase in the number of elements included in their composition, as well as the number of internal connections. These factors manifest themselves in aspects such as structural complexity, functional complexity, behavioral complexity, modeling complexity and scaling complexity.

A feature of the new «Swarm-bot» systems is that their functions, parameters, structures and behavior under the influence of internal or external factors at different time intervals of the life cycle can change, either in software or in hardware. That is, in practice the following happens, we may encounter changes in the structural dynamics of «Swarm-bot» systems of various nature. As key characteristic examples of programmable «Swarm-bot» systems with a tunable structure, one can cite:

-intelligent embedded systems for managing the operation of unmanned mobile objects;

-geographically distributed heterogeneous information and computing networks;

-internet of things.

The emergence of danger for new «Swarm-bot» systems can represent both external and internal factors that lead to the appearance of crises, accidents and disasters, of a natural-ecological or anthropogenic-social nature. In the event of such situations, ensuring the reliability, survivability,

1COLINS-2022: 6th International Conference on Computational Linguistics and Intelligent Systems, May 12-13, 2022, Gliwice, Poland
EMAIL: krivoulya@yahoo.com (G. Krivoulya); tokarev@ieee.org (V. Tokariev); ilirvi_71@ukr.net (I. Ilina); oleh.lebediev@nure.ua (O. Lebediev); vladyslav.shcherbak@nure.ua (V. Shcherbak)
ORCID: 0000-0002-6143-5628 (G. Krivoulya); 0000-0002-7143-6165 (V. Tokariev); 0000-0003-3132-7949 (I. Ilina); 0000-0001-5998-0136 (O. Lebediev); 0000-0002-5479-4438 (V. Shcherbak)

disaster tolerance of «Swarm-bot» systems as a whole and their elements separately, to perform the programmed functions at some stage of the life cycle is one of the topical strategic directions for the development of new technical systems.

The purpose of this work is to explore the possibility of using a centralized particle swarm algorithm to distribute subtasks between «s-bot» one «Swarm-bot» system to solve the main problem.

To achieve this goal, this paper solves the problem of developing an iteration algorithm based on a centralized particle swarm algorithm. When solving the task, it is necessary that «Swarm-bot» systems, regardless of the stage of the life cycle, be manageable, i.e., able to rebuild their structures, states, parameters and ways of functioning in various conditions and environments. The works show that the solution of complex problems is more effective when «Swarm-bot» systems are used as a whole, and not individual elements that are part of them, for example, separate «s-bots». Those. when using «Swarm-bot» systems, the range of action is significantly increased due to the dispersal of the «s-bots» that make up this «Swarm-bot» system over the entire surface, which significantly increases the chances of achieving the goal by redistributing the main task into subtasks between separate «s-bots» of one «Swarm-bot» systems in case of failure of some of the «s-bots».

2. Related Works

The problem of managing a group of mobile «s-bots», which must cooperatively perform a certain task, is relevant in many areas of modern life. So, this problem arises in the practice of collecting data on large objects, territories, etc. Any system consisting of individual nodes (for example, a group of central processing unit - CPU in a multiprocessor computing system) can be considered as an object of group management (collective management). It is clear that managing a team of intelligent mobile «s-bots» that are part of one «Swarm-bot» system in order to perform a specific task, on the one hand, requires the development of methods and models for managing the interaction of individual intelligent mobile «s-bots» to achieve the common goal of the «team», and on the other hand, to develop a strategy and algorithms for the implementation of these interactions by means of the team in real time and taking into account the ongoing changes in the physical unorganized environment of their functioning.

In scientific works of Kaljaev I.A., Gajduk A.R., Kapustin S.G. it is shown that one, intelligent mobile «s-bot», can't always effectively perform the task of one «Swarm-bot» system, in particular, due to a small, as a rule, range, a limited energy resource, a limited number of operations that he is able to perform, and, finally, a low probability of achieving the goal in extreme conditions associated with the possibility of failure of one, intelligent mobile «s-bot» [1,2].

A team of researchers Serkov A., Barabash O., Tverdenko H., Sobchuk V., Musienko A., Lukova-Chuiko N. published results showing that when external or internal destructive influences appear on the «Swarm-bot» systems, the most effective solution to achieve the task (increasing survivability), is the simultaneous use of a group of intelligent mobile «s-bots» included in one «Swarm-bot» system [3-5]. The use of such a complex system allows you to increase the range by dispersing intelligent mobile «s-bots» throughout the working plane, expand the set of functions that can be performed, provide a higher probability of solving the task, by increasing the functional stability of «Swarm-bot» systems [6-9].

Leading scientists of Ukraine Dodonov, A.G., Gorbachyk, O.S., Kuznietsova, M.G. published works, the material of which shows that when using intelligent mobile «s-bots» equipped with an autonomous system of movement and navigation and capable of performing certain functions, complex tasks arise, primarily related to the problem of managing such tools and organizing their collective interaction for the most effective achievement of the task [10-14]. The practice of solving problems of managing «Swarm-bot» systems shows that in the case of controlling the movement of such complex systems, the movement of an intelligent mobile «s-bots» in a system that is part of one «Swarm-bot» system is unimportant, it is necessary to determine the characteristics of the movement of the entire «Swarm-bot» system, since it forms a complex space-time structure [15-17]. Under the conditions of controlling the movement of an object as part of a group, the characteristics of the movement of an individual «s-bot» and its behavior and interaction with other «s-bot» of one «Swarm-bot» system become important [18-19]. Management is focused on ensuring the implementation of a system-wide task by a multiplicity of intelligent mobile «s-bots».

The basic properties of intelligent mobile «s-bots» today are autonomy of actions, the ability to plan and make decisions, the ability to influence the environment, intelligence based on the representation of knowledge and purposeful problem-oriented judgments, the ability to interact with information. With collective management, the quality of interaction and information exchange is characterized by orientation, selectivity, intensity, dynamism, informativeness and stability of the interaction of «s-bots» of one «Swarm-bot» system. To organize the control systems of the «Swarm-bot» system, some strategies are used that are used to control various technical, social and natural groups. One of these strategies shows that each «s-bot» of one «Swarm-bot» system decides on its own, exchanging information with other «s-bots», based on its own experience (advantages) - independent strategy formation. This approach, which is also called «packing», is implemented in the case when each «s-bot» of one «Swarm-bot» system performs its subtask and thereby makes a personal contribution to the achievement of the global task of one «Swarm-bot» system [20-23]. The subtask of one intelligent mobile «s-bot» will be relatively simple, since the task of optimizing only its actions as part of the «Swarm-bot» system is solved, without optimizing the actions of the «Swarm-bot» system. A separate intelligent mobile «s-bots» may not even have a connection with other «s-bots» of one «Swarm-bot» system but based on indirect information about changes in the state of the environment caused by the actions of other «s-bots», it can change its actions to achieve the goal. This strategy is decentralized. An important advantage of the decentralized control strategies of the «Swarm-bot» system is the increase in the overall survivability of such a system [24-26]. Since all «s-bots» are equivalent in one «Swarm-bot» system, the loss or damage of anyone «s-bots» does not lead to the loss of the entire system. And increasing the survivability of the group is achieved without additional costs, but only through the most decentralized organization of group management. Unfortunately, the decentralized group control strategies are difficult to algorithmize and, moreover, they do not guarantee the optimal solution of the group problem. But with increased requirements for the survivability of «s-bots» of one «Swarm-bot» system, decentralized control strategies should be chosen [27-29].

The emergence of a scientific direction - swarm algorithms (a swarm of particles, and an ant colony algorithm) associated with an attempt to solve complex optimization problems, makes it possible for researchers and scientists to approach many unsolved problems. The greatest difficulty in the application of swarm algorithms is their adjustment and refinement for various types of optimization problems, the selection of algorithm coefficient values to obtain high efficiency on various classes of problems, as shown in many studies, among which are the works of M. Dorigo, J. Kennedy, Yu Shi, R. Eberhart.

3. Methods

As in an unorganized physical environment - unorganized physical environment - (UPE) a collective behavior was formed, which is based on a complex interaction between the subjects of evolutionary selection, so in optimization problems, after evolutionary algorithms, a new class of algorithms appeared - swarm optimization. Today this class has formed into a group of swarm intelligence methods. In 1989, the scientific work of Gerardo Beni and Wang Jing was published, in which the authors present a new scientific direction - cellular automata. In this scientific work, the authors introduce a new term - swarm intelligence - (SI), which immediately became entrenched in the scientific community. Today, particle swarm and ant algorithms show high efficiency in solving optimization problems. Having analyzed the collective behavior in nature between individual biological populations, the authors of the scientific publication were able to develop mathematical models for technical systems and apply them to create optimization algorithms. The first mathematical model that replicated collective behavior was the model of bird behavior in a flock, created in 1986 by Craig Reynolds.

K. Reynolds, with the help of simulation modeling, managed to develop a plausible visualization of the collective behavior of birds in a flock. Further, scientists J. Kennedy and R. Eberhart in 1995 proposed an optimization algorithm for continuous nonlinear functions and called it the particle swarm optimization algorithm - (PSO). The researchers thoroughly studied the Reynolds mathematical model, as well as the modifications of this model created by that time, published in the scientific works of Heppner and Grenadier. Kennedy and R. Eberhart were able to quite plausibly model the social behavior of birds in one flock and formulate simple but basic rules for the movement of each bird in space.

Today, particle swarm optimization - PSO is one of the bionic optimization methods. Particle Swarm Optimization reflects the ability of a flock of birds, fish, and other biological populations to adapt to an unorganized physical environment, search for food resources, and avoid predators by sharing information. Particle swarm optimization optimizes a function by maintaining a population of possible solutions (each simulating a single bird) called particles and moving these particles around the solution space according to a simple formula. The movements are subject to the principle of the best position found in this space, which constantly changes when the particles find more favorable positions. Thus, an optimization method arises, for the use of which it is not necessary to know the exact gradient of the function being optimized. In the method, a group of particles is initialized randomly. Then they look for the optimal solution by performing a sequence of iterations. At each iteration, the particles update their position based on the parameters:

p_i - the best-known position of the particle i;

g - the best-known state of the swarm.

When the optimal values p_i and g are found, particle velocities and locations are updated according to the following formula:

$$v_i \leftarrow wv_i + \varphi_p r_p(p_i - x_i) + \varphi_g r_g(g - x_i), \qquad x_i \leftarrow x_i + v_i$$

where w - inertial weight; φ_p and φ_g - learning rates, usually equal to 2; r_p and r_g - random numbers on the segment [0,1].

Each bird knows its own location: the distance to food, the distance to food of other birds in its flock. At some initial point in time, the birds have some random speed, which is given by the modulus and direction, then they adjust the speed, moving towards the bird closest to the food. Such actions are described in the classical Reynolds algorithm [30]. To simulate the behavior of birds in flocks, Reynolds programmed the behavior of each of the birds separately, as well as their interaction with each other, but within the same flock. In doing so, Reynolds used the following principles. Reynolds' first principle is that every bird should strive to avoid collisions with other birds. The second principle of Reynolds is that each bird must move in the same direction as the neighboring birds. The third principle of Reynolds, each bird should try to move at the same distance from the neighboring bird. If a flock of birds is considered as a swarm of particles, then the optimal solution is found after a certain number of iterations, and at each iteration step the particle (this is one bird in the flock, and the distance to the «food» is the cost of completing tasks) updates its position, striving for its own best solution (pbest) - the local solution of the particle, and, at the same time, to the best solution among all particles of the swarm (gbest) - the global best solution. Then the correction of the speed and position of the particle will occur according to the formulas that describe the classical particle swarm algorithm [30]. At present, the particle swarm algorithm has undergone many modifications that were published at different times by researchers in this field, but the basic principles formulated by Yu. Shi and R. Eberhart.

4. Experiment

Statement of the problem of conducting an experiment, setting input parameters and imposing restrictions for one «Swarm-bot» system. The redistribution of subtasks between the «s-bot» of one «Swarm-bot» system is as follows. Let be:

$$Z = \{z_1, z_2, \dots, z_k\}, \tag{1}$$

where Z – a set whose elements are subtasks that must be solved to achieve the main task.

$$B = \{b_1, b_2, \dots, b_p\}, \tag{2}$$

where B – the set of elements of which are «s-bot», which are part of one «Swarm-bot» system.

The sets Z and B can change during the operation of «s-bot» that are part of one «Swarm-bot» system.

The following matrices are given, having the dimension m x n.

Reward Matrix:

$$A = \{a_{ij}\}, \tag{3}$$

cost matrix:

$$D = \{d_{ij}\}, \tag{4}$$

and opportunity matrix:

$$K = \{k_{ij}\}, \quad (5)$$

Table1, Table2, and Table3 respectively have the following input parameters, a_{ij} – rewards when performing the i-th subtask j-th - «s-bot», d_{ij} – resources spent by the j-th - «s-bot» on the execution of the i-th subtask, k_{ij} – the possibility of performing the i-th subtask j-th - «s-bot».

Then:

$$k_{ij} = \{0,1\}, \quad (6)$$

where $i = \{1, \dots, \mathrm{c}\}$, a $j = \{1, \dots, p\}$

It is necessary to distribute subtasks between «s-bot» - in one «Swarm-bot» system, to solve the main task Z, so that the payout is maximum:

$$X = \sum_{j=1}^{p} \sum_{i=1}^{c} (k_{ij} \times (a_{ij} - d_{ij})), \quad (7)$$

where X is win; p is the serial number of «s-bot» in one «Swarm-bot» system; c is the serial number of the subtask solved by «s-bot» of one «Swarm-bot» system; k_{ij} is the ability to perform the i-th subtask j-th «s-bot»; a_{ij} is reward when performing the i-th subtask j-th «s-bot»; d_{ij} is resources spent by the j-th «s-bot» on the execution of the i-th subtask.

Table 1

$\boldsymbol{a_{ij}}$ – rewards when performing the i-th subtask j-th «s-bot»

j - «s-bot» i - tasks	1	2	3	…	p
1	a_{11}	a_{12}	a_{13}	…	a_{1p}
2	a_{21}	a_{22}	a_{23}	…	a_{2p}
…	…	…	…	…	…
c	a_{c1}	a_{c2}	a_{c3}	…	a_{cp}

Table 2

$\boldsymbol{d_{ij}}$ – resources spent by the j-th - «s-bot» on the execution of the i-th subtask

j - «s-bot» i - tasks	1	2	3	…	p
1	d_{11}	d_{12}	d_{13}	…	d_{1p}
2	d_{21}	d_{22}	d_{23}	…	d_{2p}
…	…	…	…	…	…
c	d_{c1}	d_{c2}	d_{c3}	…	d_{cp}

Table 3

$\boldsymbol{k_{ij}}$ – the possibility of performing the i-th subtask j-th «s-bot»

j - «s-bot» i - tasks	1	2	3	…	p
1	k_{11}	k_{12}	k_{13}	…	k_{1p}
2	k_{21}	k_{22}	k_{23}	…	k_{2p}
…	…	…	…	…	…
c	k_{c1}	k_{c2}	k_{c3}	…	k_{cp}

It is convenient to present the distribution result as an array, the size of which corresponds to the number of distributed subtasks. The elements of the array are the numbers of «s-bot» in one «Swarm-bot» system participating in the distribution, and the ordinal number of the element corresponds to the number of the subtask assigned to this «s-bot». For example, the sequence [3, 1, 2] means the following, the 3rd «s-bot» will perform the 1st subtask, the 1st «s-bot» will perform the 2nd subtask, and the 2nd «s-bot» will execute the 3rd subtask. The distribution of subtasks between the «s-bot» of one «Swarm-bot» system is performed using a centralized particle swarm algorithm - CPSA. The standard deviation $\sigma[x_i]$ of a random i-th particle is calculated according to the formula:

$$\sigma[x_i] = \sqrt{D[x_i]}, \tag{8}$$

where $\sigma[x_i]$ is the standard deviation of a random i-th particle; $D[x_i]$ is the dispersion of a random i-th particle. The dispersion of a random i-th particle can be calculated by the formula:

$$D[x_i] = \sum\nolimits_{i=1}^{n} (x_i - m[x_i])^2 \times P_i, \tag{9}$$

where $D[x_i]$ is dispersion of a random i-th particle; x_i is the value of a random i-th particle among a series of n-values; $m[x_i]$ is mathematical expectation of a random i-th particle; P_i is the probability of the appearance of a random i-th particle. The mathematical expectation of a random i-th particle is calculated according to the classical formula:

$$m[x_i] = \sum\nolimits_{i=1}^{n} (x_i \times P_i), \tag{10}$$

where $m[x_i]$ is the mathematical expectation of a random i-th particle; P_i is the probability of the appearance of the i-th particle. The probability of the appearance of the i-th particle, among a series of n values, can be calculated according to the formula:

$$P_i = \frac{1}{n}, \tag{11}$$

where P_i is the probability of the appearance of the i-th particle; n is the total number of values in one sample.

5. Results

The algorithm proposed for research is iterative and consists of the following iterations:

- iteration№1. Random swarm initialization. As in any technical complex system, there are 3 basic points in the algorithm under study (initial, current and final). Based on this, we begin to consider the particle swarm algorithm at the starting point. At this moment, «s-bots» are randomly located throughout the search plane for the best solution, each of the «s-bots» at this point has an arbitrary speed and some arbitrary direction of movement;
- iteration№2. The base values of the objective function are calculated, namely the total distance between the «s-bot» in one «Swarm-bot» system and the destination points. Based on the results of the calculations, a conclusion is made about the best local and global solution;
- iteration№3. Actions are performed related to correcting the position of the particle on the plane so that it does not go beyond certain search boundaries and attempts for the best solution. Correction of the position of the particle is carried out based on the formulas of the classical particle swarm algorithm [30]:

$$v_i = v_i + u_1 \times rnd(0,1) \times (pbest - x_i) + u_2 \times rnd(0,1) \times (gbest - x_i), \tag{12}$$

$$x_i = x_i + v_1, \tag{13}$$

where v_i is the speed at some current point of an arbitrary i-th particle; x_i is position at some current point of an random i-th particle; u1 and u2 are the weight coefficients of the local and global solutions, respectively; pbest is the best solution determined by an random i-th particle (local optimum); gbest is the best solution among all particles of one set (global optimum); rnd (0, 1) is random numeric value in the range from 0 to 1;

- iteration№4. Checking the conditions for stopping the algorithm. If the number of iterations is equal to a given number, then if the specified conditions and restrictions are met, the search ends, otherwise, go to iteration№2.

Starting to solve the task, it is necessary to ensure the optimal distribution of subtasks – z between a certain number of «s-bot» in one «Swarm-bot» system to solve the main task Z. This considers the promotion of each «s-bot» included in the composition of one «Swarm-bot» system – matrix A (Table1) and the resources spent by each «s-bot» in solving its subtask – matrix D (Table2). The point of using the particle swarm algorithm is to explore the possibility of distributing subtasks between «s-bot» – one «Swarm-bot» system in such a way that each «s-bot» that is part of one «Swarm-bot» system when solving the received problem, he sought to achieve the target functional and bring as much benefit as possible so that the win – X was maximum. In the algorithm at a certain iteration, it is necessary to take into account the interests of each «s-bot» that is part of one «Swarm-bot» system according to its subtask.

For example. Let's set the composition of «Swarm-bot» system equal to 5 – «s-bot». Let's introduce restrictions that any of the subtasks can be solved by any of the five «s-bots» that are part of one «Swarm-bot» system. Table4, Table5, Table6 present the results of experimental studies of the work of the developed iteration algorithm based on the centralized particle swarm algorithm. So Table4 presents the results of experimental studies depending on the number of particles in the swarm. In the classical particle swarm algorithm, each individual bird in the flock is represented by a single particle. In the case of the «Swarm-bot» system, the particle is the solution of the problem of distributing subtasks between the «s-bots» that are part of the same «Swarm-bot» system, and the distance to the «food» is the cost of completing the subtask.

Table 4

Results of experimental studies of the work of the developed iteration algorithm based on the centralized particle swarm algorithm (depending on the number of particles)

Number of particles	Costs min	Costs max	Average costs (min+max)/2	RMS deviation, $\sigma[x_i]$
5	3.255	4.800	4.027	0,617
10	3.230	4.600	3.915	0,410
15	2.955	4.700	3.827	0,434
20	3.050	4.500	3.775	0,314
25	3.205	4.200	3.702	0,193
30	2.700	4.300	3.500	0,299
35	3.005	4.250	3.627	0,205
40	3.000	4.100	3.550	0,169

Table 5

Results of experimental studies of the work of the developed iteration algorithm based on the centralized particle swarm algorithm (depending on the number of iterations)

Number of iterations	Costs min	Costs max	Average costs (min+max)/2	RMS deviation, $\sigma[x_i]$
10	5.010	5.900	5.455	0,266
20	4.500	4.850	4.675	0,075
30	4.150	4.600	4.375	0,079
40	4.000	4.450	4.225	0,068
50	3.950	4.350	4.150	0,054
60	3.650	3.900	3.775	0,033
70	3.200	3.550	3.375	0,042
80	2.950	3.150	3.050	0,029

Table 6

Results of experimental studies of the work of the developed iteration algorithm based on the centralized particle swarm algorithm (depending on the number of subtasks)

Number of tasks	Costs min	Costs max	Average costs (min+max)/2	RMS deviation, $\sigma[x_i]$
4	7.150	7.800	7.475	0,281
5	10.000	11.150	10.575	0,459
6	13.950	14.700	14.325	0,279
7	17.000	18.200	17.600	0,419
8	20.550	22.000	21.275	0,479
9	24.150	25.400	24.775	0,392
10	27.900	29.000	28.450	0,329
11	31.450	33.300	32.375	0,531

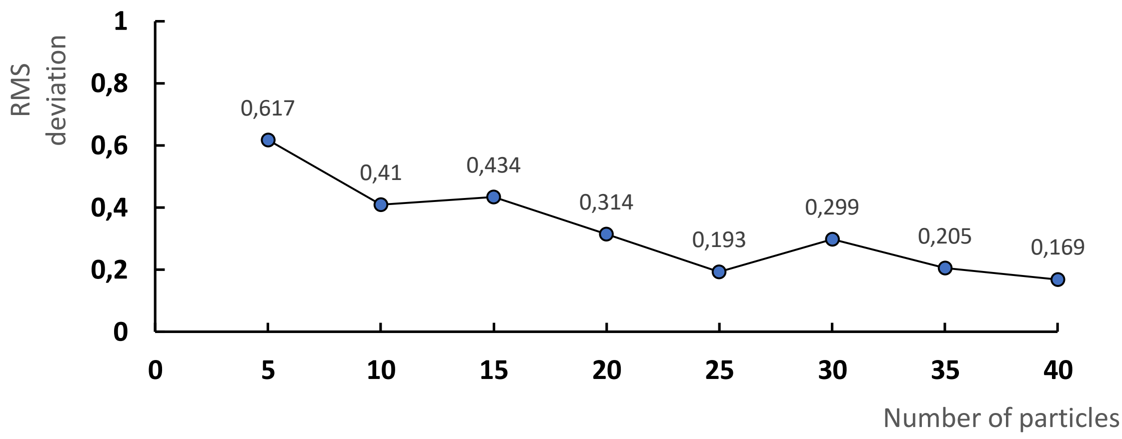

Figure 2: Graph of the number of particles

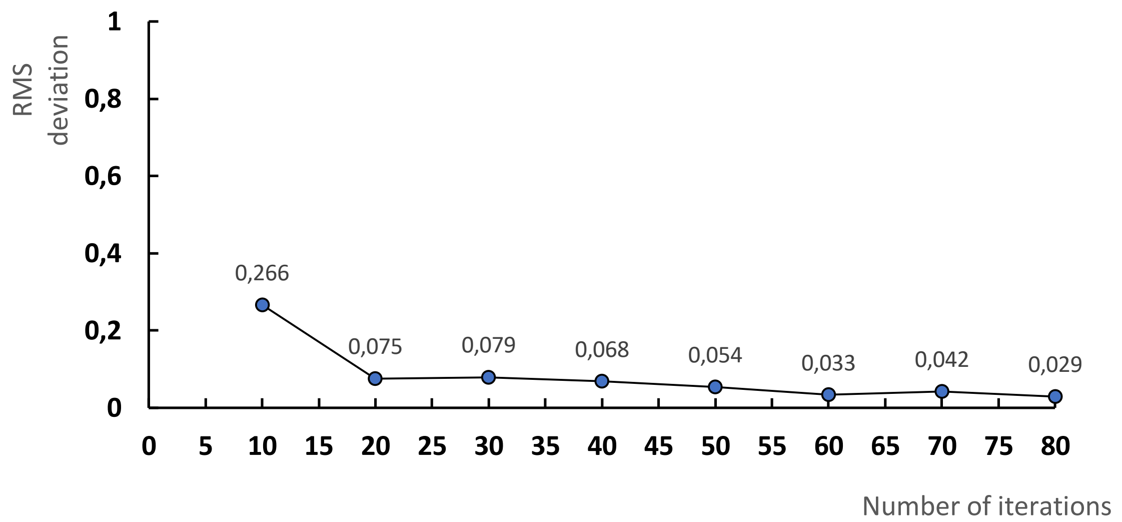

Figure 3: Iteration number graph

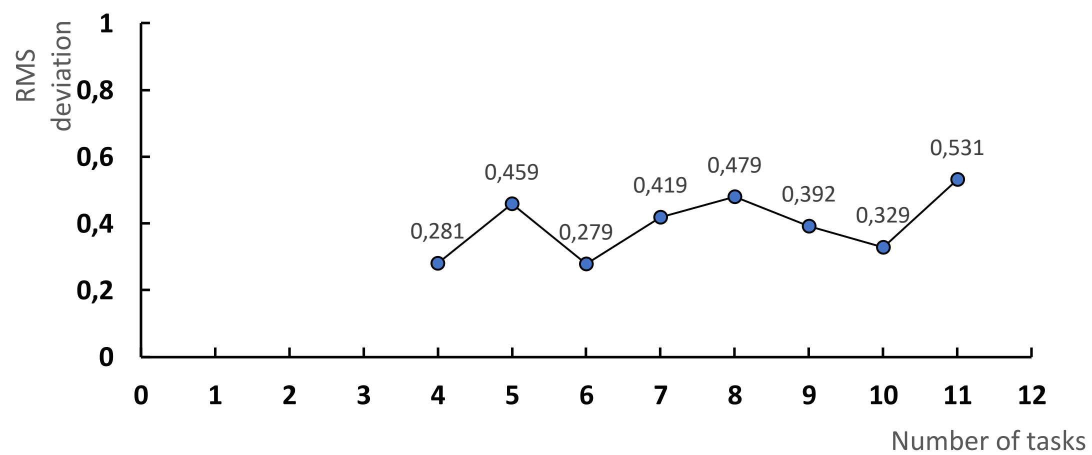

Figure 4: Task Number Graph

6. Discussions

After analyzing the data in Table 4 and Table 5, we can say that an increase in the number of iterations and the number of particles significantly improves the desired solution of distributed subtasks in order to achieve a global task. The calculated values of the standard deviation (the rightmost column of Table 4, Table 5 and Table 6) are rather small values, so it can be argued that the developed iteration algorithm has stable convergence to find the optimal solution of the problem. After analyzing the data located in Table 6, we can say the following - with an increase in the number of subtasks, to solve a global problem, the running time of the developed iteration algorithm will increase, but the standard deviation will remain small.

Based on the practical and theoretical studies carried out, a sequence of iterations based on a centralized particle swarm algorithm was developed and successfully tested:

- iteration№1. Random initialization of the «s-bot» swarm in one «Swarm-bot» system, i.e. «s-bot» are located randomly on the entire plane, which is given by arbitrary initial and final boundaries. Each «s-bot» at the initial moment of time has an arbitrary speed and direction of movement. In our case, the «Swarm-bot» system consists of a swarm of 4 «s-bots» with numbers {1, 2, 3, 4}. The main task of the «s-bot» swarm is the rescue operation of a lost group of tourists in a wooded area. The coordinates of the «s-bot» swarm and the group of tourists are known. Then the costs are determined by the distances between the «s-bot» swarm and the group of tourists;
- iteration№2. Computing the value of the objective function, i.e., finding the distance between the «s-bot» swarm and a group of tourists. Having received the input data, each «s-bot» determines the local and global optimal solutions.
- iteration№3. Actions are taken to correct the position of each «s-bot» on the ground so that it does not go beyond the specified search boundaries.
- iteration№4. Checking the conditions for stopping the work of the developed sequence of iterations. Checking the fulfillment of specified conditions. If they are satisfied, the search ends, otherwise - return to iteration№2.

During practical research, it was found that the developed iteration algorithm based on the centralized particle swarm algorithm is characterized by ease of operation, a small set of input parameters that must be set at the first iteration, sufficiently acceptable accuracy, and what is especially encouraging, the developed algorithm has a fast convergence to making the optimal decision.

7. Conclusions

In this work, the task of investigating the possibility of using a centralized particle swarm algorithm in «Swarm-bot» systems was set and successfully solved. The developed iteration algorithm has shown that it is an effective algorithm, as it allows you to find the best solution much faster. During practical research, it was found that the developed iteration algorithm based on the centralized particle swarm algorithm is characterized by ease of operation, a small set of input parameters that must be set at the first iteration, sufficiently acceptable accuracy, and what is especially encouraging, the developed algorithm has a fast convergence to making the optimal decision. Based on this, we can conclude that the use of the particle swarm algorithm with a centralized distribution of subtasks between «s-bot» in one «Swarm-bot» systems to solve the main task is appropriate and justified. It is supposed to continue research in this direction and conduct experiments on the distribution of subtasks between «s-bot» in one «Swarm-bot» systems using a genetic centralized algorithm. Next conduct a comparative analysis of the obtained values and based on the results of the analysis develop recommendations on the appropriateness of using one or another algorithm in Swarm-bot systems.

8. References

[1] Shuang L., Dong S., Changan Z. Coordinated motion planning for multiple mobile robots along designed paths with formation requirement // ASME Transactions on Mechatronics. – 2018. – Vol.16. – P. 1021–1031.

[2] Hu J., Xu J., Xie L. Cooperative search and exploration in robotic networks // Unmanned Systems. – 2017. – Vol.01. – P. 121–142.
[3] Hsieh M.A., Kumar V., Chaimowicz L. Decentralized controllers for shape generation with robotic swarms // Robotica. – 2018. – Vol.26(5). – P. 691–701.
[4] Haghighi R., Cheah C. Multi-group coordination control for robot swarms // Automatica. – 2017. – Vol.48. – P. 2526 – 2534.
[5] Gazi V. Swarms aggregation using artificial potentials and sliding mode control // Transactions on Robotics. – 2017. – Vol.21(4). – P. 1208–1214.
[6] Efrima A., Peleg D. Distributed algorithms for partitioning a swarm of autonomous mobile robots // Theoretical Computer Science. – 2019. – Vol.410. – P. 1355–1368.
[7] Churyumov G., Tokarev V., Tkachov V., Partyka S. Scenario of Interaction of the Mobile Technical Objects in the Process of Transmission of Data Streams in Conditions of Impacting the Powerful Electromagnetic Field // IEEE Second International Conference on Data Stream Mining & Processing: (DSMP)., 21-25 aug. 2018 y. – Lviv, 2018. - P.183-186.
[8] Tokariev V., Tkachov V., Ilina I., Partyka S. Implementation of combined method in constructing a trajectory for structure reconfiguration of a computer system with reconstructible structure and programmable logic. // Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security": (ITS 2019). CEUR Workshop Processing., 28 nov. 2019 y. - Kyiv, 2019. - P. 71-81.
[9] Yan X., Chen J., Sun1 D. Multilevel-based topology design and shape control of robot swarm // Automatica. – 2017. – Vol. 48. – P. 3122–3127.
[10] Stankevich L., Serebryakov S., Ivanov A. Data mining techniques for RoboCup soccer agents // Proc. AIS-ADM’2005. – Vol. 3505 LNAI, 2005. – P. 289–301.
[11] Ren W., Sorensen N. Distributed coordination architecture for multirobot formation control // Journal of Robotics and Autonomous Systems. – 2018. – Vol.56(4). – P. 324–333.
[12] Lawton J.R., Beard R.W., Young B.J. A decentralized approach to formation maneuvers // Transactions on Robotics and Automation. – 2018. – Vol.19(6). – P. 933–941.
[13] R. O’Grady, A. L. Christensen, M. Dorigo, SWARMORPH: Morphogenesis with self-assembling robots, in Morphogenetic Engineering: Toward Programmable Complex Systems, R. Doursat, H. Sayama, O. Michel, Eds. (Springer, 2012). - Chap. 2. - P. 27–60.
[14] Krivoulya G., Ilina I., Tokariev V., Shcherbak V. Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic / G. Krivoulya, V. Tokariev, I. Ilina, V. Shcherbak // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology: (PIC S&T), 06-09 oct. 2020y. - Kharkiv, 2021. - P.573 - 576.
[15] K. Smelyakov, M. Shupyliuk, V. Martovytskyi, D. Tovchyrechko and O. Ponomarenko, "Efficiency of image convolution," 2019 IEEE 8th International Conference on Advanced Optoelectronics and Lasers (CAOL), 2019, pp. 578-583, doi: 10.1109/CAOL46282.2019.9019450.
[16] Smelyakov K., Smelyakov S., Chupryna A. Advances in Spatio-Temporal Segmentation of Visual Data. Chapter 1. Adaptive Edge Detection Models and Algorithms. Series Studies in Computational Intelligence (SCI), Vol. 876. – Publisher Springer, Cham, 2020. – P. 1-51. DOI https://doi.org/10.1007/978-3-030-35480-0.
[17] K. Smelyakov, A. Chupryna, O. Bohomolov and I. Ruban, "The Neural Network Technologies Effectiveness for Face Detection," 2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP), 2020, pp. 201-205, doi: 10.1109/DSMP47368.2020.9204049.
[18] A. F. T. Winfield, J. Nembrini, Emergent swarm morphology control of wireless networked mobile robots, in Morphogenetic Engineering: Toward Programmable Complex Systems, R. Doursat, H. Sayama, O. Michel, Eds. (Springer, 2012). - Chap. 10. - P.239–271.
[19] M. H. M., Narayan, A., and Tuci, E. Cooperative object transport with a swarm of e-puck robots. Robustness and scalability of evolved collective strategies // Swarm Intell. 2017. Vol. 11. P.185–209. Doi: 10.1007/s11721-017-0135-8.
[20] Albani, D., Manoni, T., Arik, A., Nardi, D., and Trianni, V. Field coverage for weed mapping: toward experiments with a UAV swarm // in Proceedings of the 11th EAI International Conference

on Bio-inspired Information and Communications Technologies (Cham). 2019. P.1–16. Doi: 10.1007/978-3-030-24202-2_10.

[21] Arvin, F., Espinosa, J., Bird, B., West, A., Watson, S., and Lennox. An affordable open-source mobile robot for education and research // J. Intell. Robot. Syst. 2019. Vol.94 P.761–775. Doi: 10.1007/s10846-018-0866-9.

[22] Arvin, F., Watson, S., Turgut, A. E., Espinosa, J., Krajník, T., and Lennox, B. Perpetual robot swarm: long-term autonomy of mobile robots using on-the-fly inductive charging // J. Intell. Robot. Syst. 2017. Vol.92. P.1–18. Doi: 10.1007/s10846-017-0673-8.

[23] Bagnato, A., Bíró, R. K., Bonino, D., Pastrone, C., Elmenreich, W., Reiners, R., et al. Designing swarms of cyber-physical systems: the H2020 CP Swarm project: invited paper // in Proceedings of the Computing Frontiers Conference (New York, NY: Association for Computing Machinery), 2018. P.305–312. Doi: 10.1145/3075564.3077628.

[24] Barciś, A., Barciś, M., and Bettstetter, C. Robots that sync and swarm: a proof of concept in ROS 2 // in Proceedings of the International Symposium on Multi-Robot and Multi-Agent Systems (New Brunswick, NJ: IEEE), 2019. P.98–104.

[25] Blender, T., Buchner, T., Fernandez, B., Pichlmaier, B., and Schlegel, C. Managing a mobile agricultural robot swarm for a seeding task // in Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (Florence), 2018. P.6879–6886. Doi: 10.1109/IECON.2016.7793638.

[26] Dhanaraj, N., Hewitt, N., Edmonds-Estes, C., Jarman, R., Seo, J., Gunner, H., et al. Adaptable platform for interactive swarm robotics (apis): a human-swarm interaction research testbed // in Proceedings of the 19th International Conference on Advanced Robotics (Belo Horizonte), 2019. P.720–726. Doi: 10.1109/ICAR46387.2019.8981628.

[27] Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., et al. Evolution of collective behaviors for a real swarm of aquatic surface robots // PLoS ONE. 2018. Vol.11. P.151-162. Doi: 10.1371/journal.pone.0151834.

[28] Farinelli, A., Raeissi, M. M., Marchi, N., Brooks, N., and Scerri, P. Interacting with team oriented plans in multi-robot systems // Agents Multi Agent Syst. 2017. Vol.31, P.332–361. Doi: 10.1007/s10458-016-9344-6.

[30] Wang Yan, Zeng Jianchao. A survey of a multi-objective particle swarm optimization algorithm // CAAI Transactions on Intelligent Systems. 2018. Vol. 5(5). P. 377–384.