Journal of King Saud University – Computer and Information Sciences 35 (2023) 101580

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Method and algorithm for task allocation in a heterogeneous group of UAVs in a clustered field of targets

Vyacheslav Petrenko [a], Fariza Tebueva [a], Vladimir Antonov [a,*], Sergey Ryabtsev [a], Andrey Pavlov [a], Artur Sakolchik [b]

[a] North-Caucasus Federal University, st. Pushkin 1, Stavropol 355007, Russian Federation
[b] Belarusian State University, 4 Nezavisimosti Avenue, Minsk 220030, Belarus

ARTICLE INFO

Article history:
Received 18 October 2022
Revised 8 March 2023
Accepted 15 May 2023
Available online 19 May 2023

Keywords:
Multi-robotic systems
Decentralized task allocation
Swarm robotics
Group control
Task allocation
Labor division

ABSTRACT

The article presents a method for distributing tasks to agents of a heterogeneous UAV group in a cluster field of tasks, when the number of tasks exceeds the number of agents by 5–20 times. The proposed task distribution method based on a three-stage procedure for distributing agents of different specializations among task clusters, taking into account the agent value function. To evaluate the effectiveness, the method compared with the greedy task distribution algorithm, the collective plan improvement algorithm, and the consensus-based linking algorithm with local rescheduling. 2400 experiments were carried out with different group sizes and randomly generated task maps, the results of which revealed the high efficiency of the proposed method. According to the results of the study, a relationship found between the efficiency of the method depending on the concentration of the number of tasks per agent. With an increase in the specific number of tasks per agent, the task execution time improves and the indicator of the path traveled by agents worsens. With a ratio of 5–10 agents per 100 tasks, the method shows the best results in terms of the parameters of the path traveled by agents and task execution time.

1. Introduction

1.1. Problems and state of research

Robotic applications and the intensive development of microelectronics have led to the miniaturization of robots and the ability to use multi-robot systems (MRS). In this paper, a multi-robot system understood as a homogeneous (swarm) or heterogeneous group of mobile robotic agents with a decentralized control system for the joint implementation of a global task.

The main properties of MRS are scalability, communication (communication between agents within the group), coordination, cooperation (collective decision-making). Heterogeneity reflected in the difference in the set of specializations and sensory equipment of agents (Kalyaev et al., 2009; Zakiev et al., 2018).

The advantages of MRSs and swarms are high mobility, low maintenance costs, the ability to perform many tasks, as well as scalability. The UAV group is a special case of the MRS and has its own specifics, which consists in the use of the agents in the airspace.

MRSs are usually stochastic, nonlinear, so building mathematical models to test and optimize control models is difficult. The lack of methods of transition from the specific behavior of an agent to the universal behavior of a group does not allow building an effective management system for groups of robots (Chung et al., 2018).

In this regard, there is a number of problems in MRS control. One of these problems is widely known as labour division (task allocation, task assignment).

Solving the problem of task allocation is also relevant when using groups and swarms of UAVs, which are a special case of the MRS. The main tasks solved by UAVs are: survey and exploration of territories, detection of dangerous objects or places of emergency, monitoring the condition of various objects, mapping the terrain, search and rescue, etc. (Chung et al., 2018). The use of UAVs allows performing multiple homogeneous and heterogeneous tasks, including the cases with a significant excess of the number of tasks over the number of agents.

* Corresponding author.
E-mail address: Ant.vl.02@gmail.com (V. Antonov).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101580
1319-1578/

Nomenclature

Designation	Meaning
1. Section "Problems and state of research"	
k	The number of groups of tasks
C_n^k	The number of combinations of k objects from a set with n objects
t	Total calculation time
N	Total number of tasks
2. Section "The studied analogues"	
M	Total number of agents
N	Total number of tasks
D_i	Array of efficiency values of the i-th agent
$d_{i,j}$	Efficiency metric of the i-th agent for the j-th task
A	The set of agents of the group
Y_c	The value of the target functional
q_i	The i-th task
$n_{j,l} = \{0, 1\}$	The l-th target label of j-th agent
a_i	The i-th agent of the set A
Δd_j	Change in the target functional
3. Section "Mathematical formulation of the problem"	
A	The set of agents of the group
a_i	The i-th agent
n	The number of agents of the set A
Q	The set of tasks
q_i	The i-th task
s	The number of tasks of the set Q
K	The set of characteristics of agents and tasks
k_j	The j-th characteristic
u	The number of characteristics in the set K
t	Task execution time
e_i	Energy potential of agent i
$W = \{w_1; w_2; \cdots; w_k; \cdots w_c\}$	Clusters of tasks
$B = [b_1; b_2; \cdots; b_p; \cdots b_u]$	The set of base stations
$\dot{X}$	Limitation on the number of agent characteristics
$\dot{Y}$	Limitation on the number of task characteristics
4. Section "Proposed task allocation method and algorithm"	
X	Cube face size
$W = \{w_1; w_2; \cdots; w_k; \cdots w_c\}$	Clusters of tasks
$P_i = \{p_1; p_2; \cdots; p_k; p_s\}$	The set of planes of the i-th cluster
F	Efficiency matrix
p_{ij}	Efficiency metric of the i-th agent in the j-th cluster
$K_j[a_i] = \{0; 1\}$	Feature label
$K([W_i])_j$	Number of characteristics j of the cluster W_i
K	The set of characteristics
k_i	The j-th characteristic
A	The set of agents
$\lvert A \rvert$	Cardinality of the set A
n	Number of agents
S	General performance metric
$\widehat{S} = \left\{\widehat{S_1}, \widehat{S_2}, \cdots, \widehat{S_n}\right\}$	The set of local maxima of the metric S
$\bar{L} = \left\{\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_k\right\}$	The set of agents unallocated across clusters
$L = \{a_{k_1}, a_{k_2}, \cdots, a_{k_l}\}$	The set of agents distributed across clusters
$\bar{Q}_i = \left\{\bar{q}_1{}^i, \bar{q}_2{}^i, \cdots, \bar{q}_s{}^i\right\}$	The set of tasks of the i-th cluser
$\widetilde{Q}$	The set of unallocated tasks
$R = \{R_1, R_2, \cdots, R_n\}$	The set of preliminary distances between tasks
$\bar{A} = \{a_{1_k}, a_{2_l}, \cdots, a_{n_s}\}$	The set of agents sorted by R_i in ascending order
$\widetilde{A_j} = \left\{a_{1_j}, a_{2_j}, \cdots, a_{s_j}\right\}$	The set of agents with the characteristic k_j
$V_j = \left\{v_{1_j}, v_{2_j}, \cdots, v_{s_j}\right\}$	The set of numbers of tasks of category k_j
$T\left[K_{j_{\widetilde{Q}}}\right]$	The number of tasks with the characteristics k_j of the set $\widetilde{Q}$
5. Section "Method of simulated annealing"	
t_{min}	Lower temperature limit
t_{max}	Upper temperature limit
t_v	Initial temperature
s_v	The set representing the order of tasks execution
$F(s_v)$	New state generation function
$\varphi(s_v)$	Power consumption function
$P(\Delta\varphi)$	Probability function
v	Number of iterations
6. Section "Evaluation of computational complexity"	
N	Number of tasks
M	Number of agents
N_i	Number of tasks in i-th cluster
v	Number of iterations

This article is devoted to the task allocation (division of labor, solving the assignment problem) in a group of UAVs, provided that the number of heterogeneous tasks significantly exceeds the number of agents by 5–20 times.

The main problem of the assignment, taking into account heterogeneous specializations of tasks and agents, is the need to iterate over NP solutions of the type "agent – tasks". The problem lies in the complexity of evaluating $C_n^k = \frac{n!}{(n-k)!k!}$ combinations, where n is the number of tasks and k is the number of clusters the tasks grouped into. The maximum of such a function is achieved when $k = \lceil \frac{n}{2} \rceil$. Let assume that the calculations are performed using a hypothetical computer processing one billion "agent – task" pairs per second. Then to calculate combinations of 36 tasks for 18 groups in the worst case, $C_{36}^{18} = 9075135300$ actions are needed (about 9 s of calculations on assumed machine), for combinations of 40 tasks for 20 groups the calculation time is approximately 138 s.

The relationship between the calculation time and the number of characteristics x is expressed by the following formula:

$$t * 10^9 = \left(\frac{\frac{N}{x}!}{\frac{N}{2x}!\left(\frac{N}{x} - \frac{N}{2x}\right)!}\right)^x \tag{1}$$

where t is the total calculation time, N is the total number of tasks.

The task allocation between the agents of UAV swarm or the agents of MRS is an urgent problem for researchers. Many well-known scientists have proposed methods and algorithms for solving it.

The analysis of the paper (Pshikhopov et al., 2015) shows a large variety of theoretical methods for solving this problem, especially for the case of equal numbers of agents and tasks. Heuristic algorithms (Kowalczyk, 2002; Mathew et al., 2015), analytical algorithms (Zavlanos et al., 2008; Notomista et al., 2019), market economy models (Zavlanos et al., 2008; Bertsekas and Castanon, 1991; Luo et al., 2015), methods of potential fields (Zavlanos and

Pappas, 2008; Zavlanos and Pappas, 2007), probabilistic and random algorithms (Berman et al., 2009; Liu et al., 2020), methods based on machine learning and artificial neural networks (Mouton et al., 2011; Zhao et al., 2021) can be distinguished by popularity, fuzzy logic methods (Mukhedkar and Naik, 2013; Wei et al., 2021), ant algorithms (Yuan et al., 2008; Payton et al., 2001; Payton et al., 2005; Oliveira et al., 2017; Liao et al., 2014; Brutschy et al., 2012), dynamic and integer programming methods (Murphy, 1999; Sikanen, 2008; Yu and LaValle, 2016), genetic algorithms (Shimaa et al., 2006; Patel et al., 2020; Soleimanpour-Moghadam and Nezamabadi-Pour, 2020), blockchain and cloud computing (Husheng et al., 2021; Msala et al., 2019), mixed algorithms (Zhang et al., 2012), particle swarm optimization (Kong et al., 2019; Wei et al., 2020), etc.

The paper (Liao et al., 2014) considers the application of iterative algorithms of task allocation, including for the case of exceeding the number of tasks over the number of agents. Ensuring high convergence of iterative algorithms is an advantage of their application.

Recent studies on the allocation of tasks in heterogeneous groups of robotic systems are presented in (Chen et al., 2022; Cui et al., 2018; Majeed and Lee, 2018; Buckman et al., 2019; Zhang et al., 2020; Shang, 2021; Liu et al., 2019; Shang, 1844). In the publications studied, the emphasis placed on various factors: dynamic changes in tasks, the presence of obstacles in the environment, the allocation of tasks in conditions of environmental uncertainty or information security incidents in the process of reaching consensus during voting.

In (Chen et al., 2022; Buckman et al., 2019; Zhang et al., 2020), modifications of the consensus-based decentralized task allocation algorithm (CBBA) considered, which is a close analogue to the proposed method in terms of implementing methods. The specificity of the solutions under consideration lies in the modification of CBBA for the tasks of a dynamic environment with a partial reallocation of tasks.

In (Cui et al., 2018), the allocation of tasks carried out under conditions of environmental uncertainty, while in (Majeed and Lee, 2018) there fixed deterministic obstacles in the environment. In the papers (Shang, 2021; Shang, 1844), the allocation of tasks is carried out in the conditions of Byzantine generals, which complicates the procedure of decision-making based on consensus.

In (Romeijn and Romero Morales, 2000), a greedy algorithm for the allocation of tasks is considered. The greedy algorithm is a universal solution to the assignment problem, providing high convergence of the solution and ease of implementation in practice. However, these positive aspects compensated by such a disadvantage as low efficiency of task allocation. Under the effectiveness of performing multiple tasks by the MRS group, it is possible to allocate the total distance traveled by agents and the execution time of the global task and all local tasks.

This article differs from those presented by considering the case of the allocation of tasks in a group of UAVs, provided that the number of heterogeneous tasks significantly exceeds the number of agents.

Earlier in (Petrenko et al., 2022; Petrenko et al., 2020), a method of task allocation (division of labor) in a swarm of UAVs monitoring a dynamic emergency zone was proposed. This paper presents an adaptation of the presented method for a heterogeneous group of UAVs and a wide range of experimental studies of the effectiveness of the developed method.

1.2. Purpose and objectives

The purpose of this article is to develop a method for allocating tasks in a heterogeneous group of UAVs in a clustered field of tasks when the number of tasks is significantly higher than the number of agents, and to demonstrate the efficiency of the proposed solution in comparison with the analogues. The proposed method designated as a method for allocating tasks with clustering and dissection.

The conditions of the problem to solved are as follows. There is a heterogeneous group of UAVs and a field of heterogeneous tasks. The types and specializations of tasks and agents are identical. The communication system between agents implemented in the form of a fully connected graph. The radius of communication between agents exceeds the geometric parameters of the task field. The task considered completed if the agent reaches the task location, i.e. the task execution time is zero. The purpose of agents (the global task of agents) is to allocate and complete all tasks in the field.

The article sets and solves the following research tasks:

- mathematical formulation of the problem;
- description of the proposed method;
- description of the modification of the simulated annealing method;
- building a software simulation;
- comparison of the efficiency of the proposed method with the analogues.

1.3. The studied analogues

In this paper, as analogues are considered:

- greedy task allocation algorithm and modification due to versatility (Romeijn and Romero Morales, 2000) (2000 г.);
- the method of collective improvement of the plan and modification in connection with the use of iterative algorithms of high convergence (Liao et al., 2014) (2009 г.);
- consensus-based bundle algorithm with local replanning for heterogeneous multi-AV system due to the uniformity of the problem solved in this study (Chen et al., 2022) (2022 г.).

1.3.1. Greedy algorithm (GrA)

Each agent is an independent computing system. The algorithm consists of 4 stages.

Stage 1. Task selection. Each agent of the group $a_i, i = \overline{1,N}$ selects the nearest available task $q_i, i = \overline{1,N}$ from the general list of tasks with similar characteristics.

Stage 2. Broadcasting of information. The agent notifies other agents about the choice of a specific task.

Stage 3. Completing the task. The agent proceeds to perform the selected task.

Stage 4. Completion of the task. The agent transmits information about the completion of the task to the other agents and proceeds to stage 1.

The results of the algorithm given in the section «Results and discussion».

1.3.2. Collective plan improvement (PCIA)

The algorithm of the collective improvement of the plan consists of 3 stages.

Stage 1. Formation of the evaluation matrix. Each agent a_i generates a one-dimensional array D_i of N performance ratings. Each agent generates performance evaluations $d_{ij}, j = \overline{1,N}$ and transfer the array D_i to other agents of the group. At the beginning of the procedure, all elements of the D_i are zeroed. As a result, each agent has a two-dimensional efficiency matrix of the size $M \times N$.

Allocation of the task among the agents of the group in such a way as to provide maximum functionality:

$$Y_c = \sum_{j=1}^{N} \sum_{l=1}^{M} d_{j,l} * n_{j,i} \to max \tag{2}$$

under restrictions

$$\sum_{l=1}^{M} n_{j,l} = 1,\ j = 1,\bar{N}, \qquad \sum_{j=1}^{N} n_{j,i} \leq 1,\ l = 1,\bar{M}, \tag{3}$$

where $d_{j,l} \geq 0$ – relative assessment of the effectiveness of achieving j agent l task.

Stage 2. Analysis of the evaluation matrix. Each agent a_i, $i = 1,\bar{N}$ analyzes its matrix and finds the maximum value of the $d_{i,l}$. If the maximum values have several estimates, then one with a lower l value selected. The value of the l index indicates the number of the candidate task to select by this agent. The estimate $d_{ih}(h \neq l)$ found, which has the value closest to the maximum, and the value $\Delta d_j = \max_{i=1} d_{il} - \max_{h=1,h\neq l} d_{ih} \left(j = 1,\bar{N}\right)$ calculated that shows how much the functional can change if agent a_i chooses another task. The selected task with the number i_j assigned to the agent if the value $d_j^{\sum} = d_{j,i_j} + \Delta d_j$ is maximum. If several agents have the same maximum values of $d_j^{\sum}$, then the task is assigned to the agent with the largest Δd_j. The agents that have not selected tasks collectively allocate the tasks that not occupied. Either stage 2 repeats until all tasks occupied or all agents have selected their tasks.

1.3.3. Collective improvement of the plan with simulated annealing (PCIASAM)

In this article, proposed an extension of the method of collective plan improvement. Inserted an additional stage in the formation of tasks sequence. The sequence of tasks is determined by a simple iteration when $q_{j_k} < 10$. If $q_{j_k} > 10$ the method of simulated annealing is used to find the shortest way to complete tasks. When assigning a final task sequence to an agent, the agent proceeds to perform the assigned tasks.

A mathematical description of the simulated annealing method given in section II C.

The algorithm under consideration and its improved version have 100% convergence of the solution, thus, the algorithm of collective improvement of the plan chosen as an analogue.

1.3.4. Greedy algorithm with simulated annealing (GrASAM)

We propose an improvement of the greedy algorithm. The improvement involves the formation of a sequence of tasks from a pre-formed sequence for each agent.

Each agent is an independent computing system. The algorithm consists of 3 stages.

Stage 1. Task selection. Each agent of the $a_i, i = 1,\bar{N}$ group selects the nearest available task $q_i, i = 1,\bar{N}$.

Stage 2. Broadcasting of information. The agent notifies other agents about the choice of a specific task. Stage 1 repeats as long as there are available tasks.

Stage 3. Formation of the task sequence. The agent forms a sequence of tasks. The sequence of tasks is determined by a simple iteration when $q_{j_k} < 10$. If $q_{j_k} > 10$ the method of simulated annealing is used to find the shortest way to complete tasks. When assigning a final task sequence to an agent, the agent proceeds to perform the assigned tasks.

Consensus-based bundle algorithm with local replanning (CBBA-LR).

From the general list of tasks, the task T closest to the agents' base point selected. Task T is allocated among the agents in 4 stages.

Stage 1. Formation of agent efficiency metrics. Agents generate performance metrics based on the distance to task T, prevalence, and feasibility.

Stage 2. Formation of lists of recalled tasks. Each agent generates a list of recalled tasks. A task j_i from the agent's task execution sequence falls into the recalled list based on the criterion of averaging the agents' traveled path and proximity to task T.

Stage 3. Redistributing agent tasks. Tasks included in the list of recalled tasks of each agent a_i excluded from the sequence of tasks of agent a_i and form a set of $J = \{j_1, j_2, \cdots, j_s, \cdots\}$ excluded tasks.

Stage 4. The set of tasks J and task T redistributed between the agents.

Agents update task sequences.

Let consider the proposed method of task allocation in a heterogeneous group of UAVs in a clustered field of tasks.

2. Method and algorithm for task allocation in a heterogeneous group of UAVs in a clustered field of tasks

2.1. Mathematical formulation of the problem

Let there be n agents a_i of the set A, tasks q_j of the set Q, and u characteristics k_j of the set K. Each agent of the set A has at most X characteristics of the set K. Each task of the set M has at most Y characteristics of the set K. The task allocation for each agent a_i consists in performing a certain number of tasks q_j, taking into account the match of agent a_i characteristics to the characteristics of tasks m_j in such a way that all tasks of the set Q are completed in time t if the agents have sufficient energy potential e_i. The task field shown schematically in Fig. 1.

Let consider the mathematical formulation of the problem. The set of agents a_i of the UAV group is denoted as $A = [a_1; a_2; \cdots; a_i; \cdots a_n]$:

$$a_i = [x_i; y_i; z_i; e_i] \tag{4}$$

where $x_i; y_i; z_i$ is current coordinates of the agent a_i; e_i is energy potential of the agent a_i.

The set of subtasks q_j (further on referred to as "tasks") of the global task $Q = [q_1; q_2; \cdots; q_j; \cdots q_m]$ presented as:

$$q_j = [x_j; y_j; z_j; e_j] \tag{5}$$

where $x_j; y_j; z_j$ are task coordinates; e_j is energy reserve of the task.

Clusters of tasks formed according to the geometric characteristics of the environment are represented as $W = \{w_1; w_2; \cdots; w_k; \cdots w_c\}$, where k is cluster number; c is total number of clusters. Clustering of the task field performed by dividing the zone into equal parts in volume. The number of clusters is equal to the number of agents of the set A. Task cluster w_k after splitting includes a certain number of tasks q_i.

The set of characteristics of tasks k_j is represented as $K = [k_1; k_2; \cdots; k_g]$.

The function e_{ij} corresponds to the energy costs of the agent a_i to move to the task q_j.

$B = [b_1; b_2; \cdots; b_p; \cdots b_u]$ is the set of base stations, where p is the number of the base station; u is the number of stations $(u \geq 1)$. Each base station characterized by coordinates:

$$b_p = [x_p; y_p; z_p] \tag{6}$$

The result of the task allocation method is a mapping R, that matches each agent a_i with a unique task $q_j \in Q$, return to base action $b_p \in B$, or a waiting task $\varnothing$:

$$R : A \to Q \cup B \cup \varnothing. \tag{7}$$

The global task Q is considered completed (condition F), if the current energy reserve of the tasks is equal to zero $(e_j = 0)$, provided that all the agents $a_i \in A$ have returned to the home station:

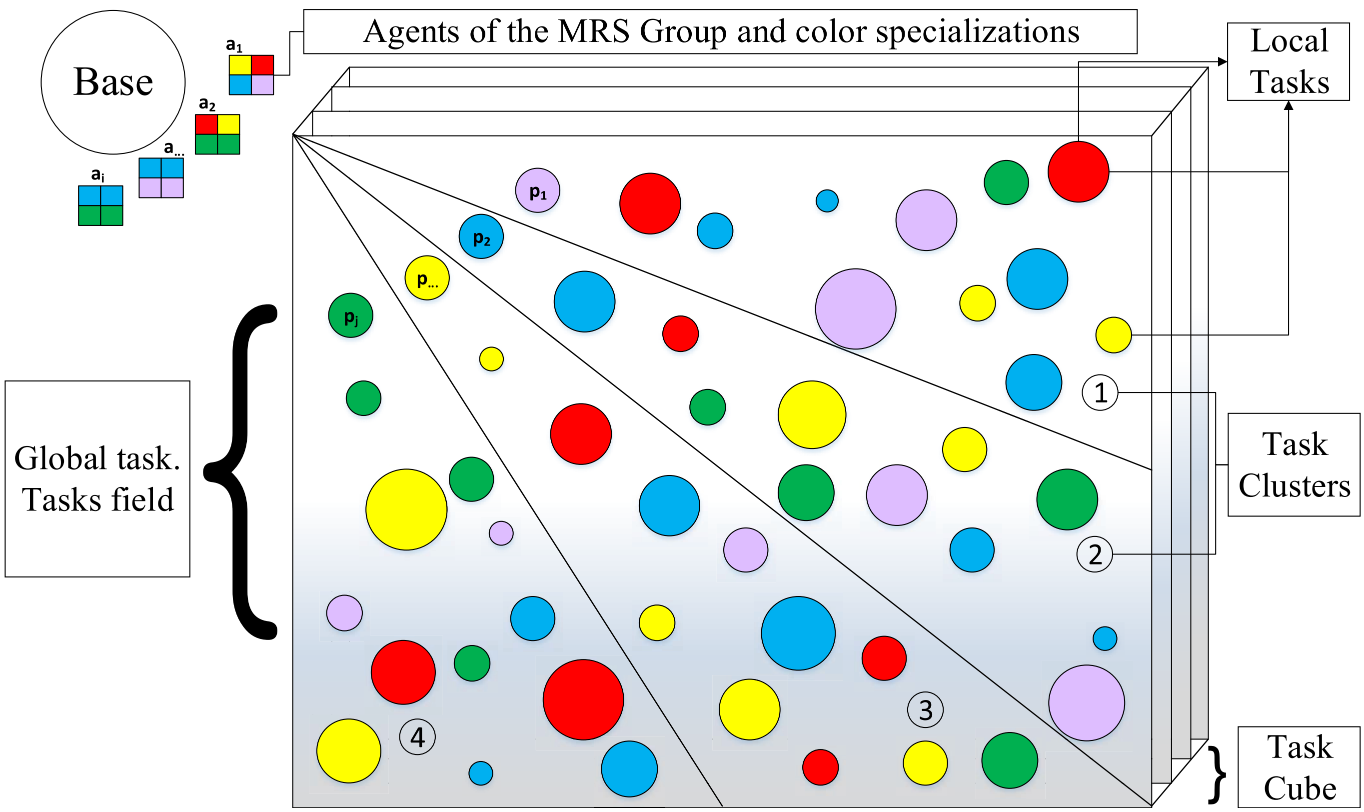

Fig. 1. The input data scheme of the proposed method for task allocation in a group of UAVs.

$$F : Q = \sum_{k=1}^{K} e_k \to 0;\ \forall a_i \in A\,R(a_i) \in B. \tag{8}$$

2.2. Proposed task allocation method and algorithm

In the proposed method, considered a particular case, when the base station is only one ($u = 1$), which is a typical case of using groups of UAVs during training flights.

To solve the proposed problem, it is proposed to use an algorithm of task allocation with collective decision-making based on majority criteria in a clustered field of task. Algorithms based on collective decision-making have high convergence of solutions, low computational complexity, the ability to search for a suboptimal solution and have great potential for further development. The possibility of modifying the method for other conditions of the problem is an important condition imposed on the method. The development of the current research suggests the presence of a "flexible" mathematical apparatus for solving the problem of division of labor.

The task allocation method consists of 5 steps.

Step 1. Selecting clusters. The input data for the agents a_i are the coordinates of the agent launch center b_p and the coordinates of the tasks q_k. Additionally, agents a_i exchange their own coordinates $a_i = [x_i; y_i; z_i]$.

Regardless of the variant of the task allocation method, each agent calculates the lengths of vectors between the centers of mass of the task clusters and the agent's base station:

$$L_{w_i} = \sqrt{\left(w_{i_x} - b_x\right)^2 + \left(w_{i_y} - b_y\right)^2 + \left(w_{i_z} - b_z\right)^2} \tag{9}$$

where i is cluster index, b_x, b_y, b_z are coordinates of the base station, $w_{ix,y,z}$ are coordinates of i-th cluster.

Agents collectively divide the task field into clusters. The initial field reduced to a cube with minimal size so that all the tasks fit in. The cube edge length is denoted as X. Next, the cube divided into clusters along the edge with coordinates $[(0,0,0);(0,0,X)]$ into parts of the same volume, the number of which is equal to the total number of agents. Each cluster is defined by a number (index) and is bounded by planes $P_i = \{p_1; p_2; \cdots; p_k; p_s\}$.

The task $q_j = [x_j; y_j; z_j; e_j]$ belongs to a cluster W_i, if bounded by planes P_i.

For each of the resulting clusters, the distance to the base station is calculated:

$$L_{W_i} = \sqrt{\left(q_{ix} - b_x\right)^2 + \left(q_{iy} - b_y\right)^2 + \left(q_{iz} - b_z\right)^2} \tag{10}$$

where i is cluster index, b_x, b_y, b_z are coordinates of the base station, $q_{ix,y,z}$ are coordinates of the i-th cluster.

Step 2. Cluster analysis. The cluster W_i received for evaluation by a group of agents.

The agents calculate the efficiency metrics of tasks in the cluster and enter into the individual collective decision-making protocol, represented by the efficiency matrix F:

$$F = \begin{array}{c|cccc} & w_1 & w_2 & \cdots & w_c \\ \hline a_1 & p_{11} & p_{12} & \cdots & p_{1c} \\ a_2 & p_{21} & p_{22} & \cdots & p_{2c} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_i & p_{i1} & p_{i2} & \cdots & p_{ic} \end{array} \tag{11}$$

The decision-making protocols are identical for all of the agents. The efficiency metric p_{ij} is determined based on the assessment of the agent's a_i ability to perform tasks in the cluster w_j as follows. The value $score_{ij}$ calculated as:

$$score_{ij} = p_{ij} = K_1[a_i]\frac{\mathrm{K}([W_j])_1}{K_1} + K_2[a_i]\frac{\mathrm{K}([W_j])_2}{K_2} + \cdots + K_g[a_i]\frac{\mathrm{K}([W_j])_g}{K_g} \quad (12)$$

where i is index of agent, j is index of cluster, $i = \overline{1,n}$, $j = \overline{1,c}$, where n is the number of UAVs, c is the number of clusters.

Value $K_j[a_i] = \begin{cases} 1, iftheagenta_ihasthecharacteristick_j, \\ 0, otherwise. \end{cases}$ displays the presence or absence of a characteristic k_j from the agent a_i. The value K_j displays the total number of tasks of characteristics k_j. The value $K([W_i])_j$ displays the number of tasks of characteristics k_j in the cluster W_i.

In this case, the agent a_i, with the set of characteristics $\{1, 2, 3\}$, of cluster W_j with the tasks of characteristics {1,2,4} of quantity {3,2,3} respectively, total number of tasks for each characteristic {10,2,12,20}, corresponds to the final metric $p_{ij} = 1 * \frac{3}{10} + 1 * \frac{2}{2} + 0 * \frac{0}{12} + 0 * \frac{3}{20} = 1$.

Step 3. Collective decision-making. Agents begin the procedure of collective decision-making. The procedure of collective decision-making uses the majority principle and includes 3 rounds.

In the first round, the agents a_i of the UAV group start decision-making protocols, in which each agent sets its own metrics p_{ij} for tasks in the cluster. The protocol considered filled if it contains the number of metrics p_{ij} equal to the product of the number of agents and the number of clusters.

In the second round, the total metric calculated which displays the optimal value of the overall performance metric:

$$S = \max_i S_i = \max_i \sum_j p_{ij} \quad (13)$$

The difficulty of calculating S is $O(n!)$.

If $|A| \leq 10$, calculations are carried out in parts. Each agent a_i, depending on its index i, searches for a local maximum $\widehat{S_i}$ on the interval $[i-1, i]$. As a result, each agent has a set of local maxima found by all agents $\widehat{S} = \left\{\widehat{S_1}, \widehat{S_2}, \cdots, \widehat{S_n}\right\}$ and selects the global maximum S. Thus, each agent will perform $(n-1)! \leq 36288$ computational operations.

If $|A| > 10$, calculations of distances between tasks are carried out in parts using the method of simulated annealing.

In the third round, agents check their records in the decision-making protocol. If the highest value of the product criterion does not relate to the index of the agent, then the agent refuses to perform tasks in the cluster. Else, the agent assigned to the cluster.

It is possible that the set of agents $\bar{L} = \{\bar{a}_1, \bar{a}_2, \cdots, \bar{a}_k\}$ not allocated across all clusters $\bar{L} \neq \varnothing$ in the case of uneven allocation of task-characteristics in the original field. Then a set $L = \{a_{k_1}, a_{k_2}, \cdots, a_{k_l}\}, L \cup \bar{L} \subseteq A$ of agents formed between which tasks will be further allocated.

Step 4. Task allocation. The process includes 2 rounds.

In the first round, agents form personal task lists. Each agent of set A forms a set $\bar{Q}_i = \{\bar{q}_1{}^i, \bar{q}_2{}^i, \cdots, \bar{q}_s{}^i\}$, consisting of cluster W_i tasks, provided that the characteristics of the tasks correspond to the parameters of the agents. Let denote by $K_j[a_i] = 1$, $j \in \overline{H[\bar{q}_1{}^i], H[\bar{q}_i{}^i]}$ the set of characteristics of the task $\bar{q}_i$ from cluster W_i.

In the second round, the agents allocate the tasks unallocated after the first round. Each agent a_i of the set A calculates the preliminary distance R_i of the task sequence $\bar{Q}_i$ by simulated annealing when the number of tasks is more than 10 or by brute force when the number of tasks is less than 10, then the agents exchange the values of these distances. As a result, each agent generates a set of distance values $R = \{R_1, R_2, \cdots, R_n\}$.

The sequence of tasks in the cluster is determined by a simple iteration at $q_{j_k} < 10$. If $q_{j_k} > 10$, then the method of simulated annealing is used to find the shortest way to complete tasks.

Tasks of the set $\widetilde{Q} = \left(Qleft\{\bar{Q}_1 \cup \bar{Q}_2 \cup \cdots \cup \bar{Q}_{|L|}\right\}) \neq \varnothing$ allocated by averaging the distance traveled between tasks by each agent as follows:

1) agents are sorted in ascending order of R_i, a set of agents $\bar{A} = \{a_{1_k}, a_{2_l}, \cdots, a_{n_s}\}$ is formed;
2) for each characteristic of the k_j a set of agents is defined:

$$\widetilde{A_j} = \left\{a_{1_j}, a_{2_j}, \cdots, a_{s_j}\right\},\ K_j\left[a_{i_j}\right] = 1, \left|\widetilde{A_j}\right| = \sum_i K_j[a_i] \quad (14)$$

3) for the agents of each set $\widetilde{A_j}$, a set $V_j = \left\{v_{1_j}, v_{2_j}, \cdots, v_{s_j}\right\}$ is formed, displaying the number of tasks of category k_j:

$$v_{i_j} = \frac{R\left[a_{i_j}\right] T\left[K_{j_{\widetilde{Q}}}\right]}{\sum_i R\left[a_{i_j}\right]} \quad (15)$$

where $T\left[K_{j_{\widetilde{Q}}}\right]$ is the number of tasks having the characteristics k_j, of the set $\widetilde{Q}$, the agents of the set $\widetilde{A_j}$ are sorted by R_i in descending order;
4) each agent of the set $\widetilde{A_j}$ replenishes the set $\bar{Q}_i$ with the nearest tasks v_{i_j}.

Step 5. Performing tasks from the generated lists.

Each agent of the set A forms a final sequence of tasks using the set $\bar{Q}_i$ to fly around, taking into account the return to the home point. The sequence of tasks determined by a simple iteration at $q_{j_k} < 10$ if $q_{j_k} > 10$ then the method of simulated annealing used to find the shortest way to complete tasks. When assigning a final task sequence to an agent, the agent proceeds to perform a flight task.

A generalized algorithm of task allocation between agents of a heterogeneous group of UAVs in a clustered field of tasks shown in Fig. 2.

2.3. Method of simulated annealing

Simulation of annealing in iterative problems used to approximate the suboptimal global minimum of functions with a large number of free variables.

The algorithm of the simulated annealing method is probabilistic and shows good results in practice when solving *NP*-complete problems. The simulated annealing allows reducing the average total traveled distance by 45% in less than 1500 iterations in comparison with the worst initial value when solving the traveling salesman problem.

Let S be the set of states of the system, which in the physical sense reflects the energy consumption function φ of agent a_i for moving through tasks q_j in the sequence w_k. The energy consumption function φ calculated by the agent based on its characteristics, state and environmental parameters, i.e. the amount of energy that the agent will spend on moving through the tasks q_j of the sequence w_k in the generated sequence of visiting tasks.

The function F, based on the initial state s_v (where v is the iteration step) generates a new candidate state s_{v+1}, into which the

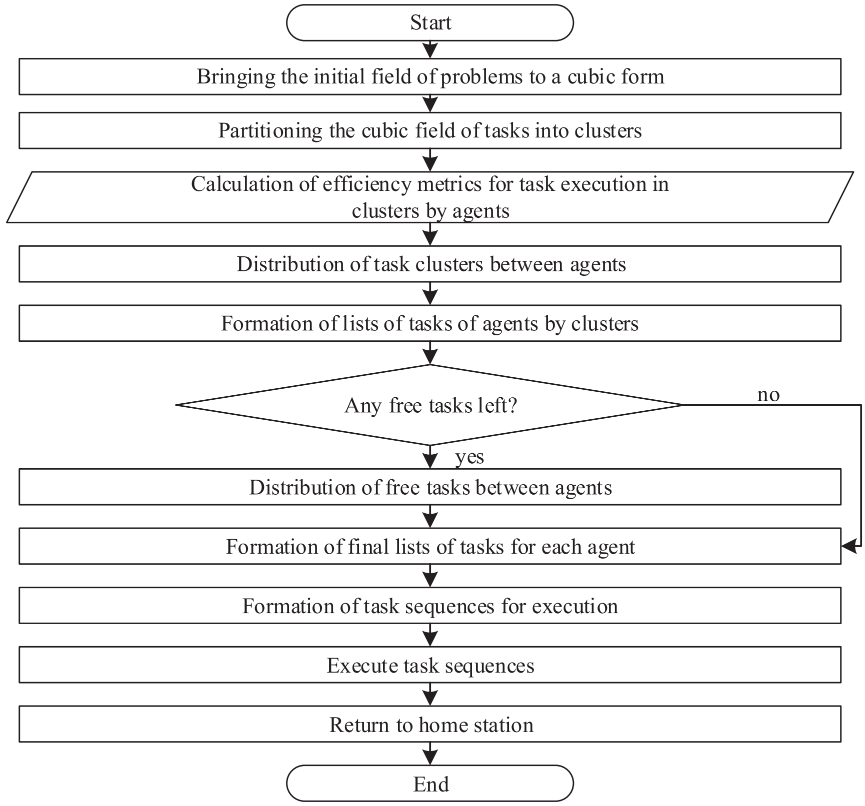

Fig. 2. A flowchart of the proposed algorithm for task allocation.

system can move, or it can discard depending on t_v – system state temperatures. Here is the algorithm of the method.

The lower temperature limit is set $t_{min} = 0$ for a state S_o;
A random state is applied to the input s_v (set the order of visiting tasks) with the initial temperature t_v = 100;
$t_v = t_{max} = 100$;
While $t_v > t_{min}$:
$s_{v+1} = F(s_v)$– starting the function of generating a new system state;
$\Delta\varphi = \varphi(s_{v+1}) - \varphi(s_v)$;
If $\Delta\varphi \le 0$, then $t_{v+1} = t_v \frac{\varphi(s_v)}{\varphi\left(s_{v+1}\right)}$;
If $\Delta\varphi \le 0$ then the temperature drops: $t_{max} = t_{v+1}$. A new iteration is repeated, where the state is fed to the input s_{v+1} and $t_{max} = t_{v+1}$.
If $\Delta\varphi > 0$ then a new iteration is carried out with probability:

$$P(\Delta\varphi) = exp^{\frac{-v}{t_{max} * \frac{\varphi\left(s_{v+1}\right)}{\varphi(s_v)}}} \tag{16}$$

For the effective operation of the method, restrictions on the number of iterations v additionally introduced.

2.4. Software simulation

Mathematical modeling carried out in C++ (Task-Allocation, 2023). Microsoft Visual Studio 2019 software, ISO C++ 14 standard, used as an integrated development environment. A graphical description of the algorithm given in Fig. 3.

To evaluate the effectiveness of the proposed solutions, conducted 2400 computational experiments conducted. Six methods implemented in the simulation: the proposed method is a method of task allocation with clustering and dissection. (DFCM) and 5 analogues: greedy algorithm (GrA), collective plan improvement (PCIA), collective plan improvement with simulated annealing (PCIASAM), greedy algorithm with simulated annealing (GrASAM), consensus-based layering algorithm with local rescheduling (CBBA-LR).

In the simulation, 100 uniformly allocated tasks generated on the map, and groups of 5, 10, 15 and 20 agents performed tasks in clusters. The results summed up for each generated map of 100x100x100 conditional units. In total, 100 maps generated for one set of agents and clusters. The model contains one home point. The characteristics of agents and tasks generated randomly for each map. Each task and agent had at least one and no more than three characteristics out of five possible, respectively. To evaluate the efficiency of task allocation, the distance traveled by all agents and the execution time measured.

3. Results and discussion

3.1. Performance evaluation

The results of experiments presented in Figs. 4, 5, where Fig. 4 shows the distance traveled by agents when performing 100 tasks, and Fig. 5 shows the execution time of global task.

In Fig. 4 and Fig. 5 the performance of the studied methods is shown: method of task allocation with clustering and dissection (DFCM); greedy algorithm (GrA), greedy algorithm with simulated annealing (GrASAM), the algorithm of collective improvement of the plan (PCIA), the algorithm of collective improvement of the plan with simulated annealing (PCIASAM), consensus-based bundle algorithm with local replanning (CBBA-LR). The number to the right of the method designation means the number of agents in the group when performing 100 tasks indicated on the abscissa axis. The ordinate axis reflects the conventional units of distance and task execution time. The box plots of the results shown in Figs. 6-7.

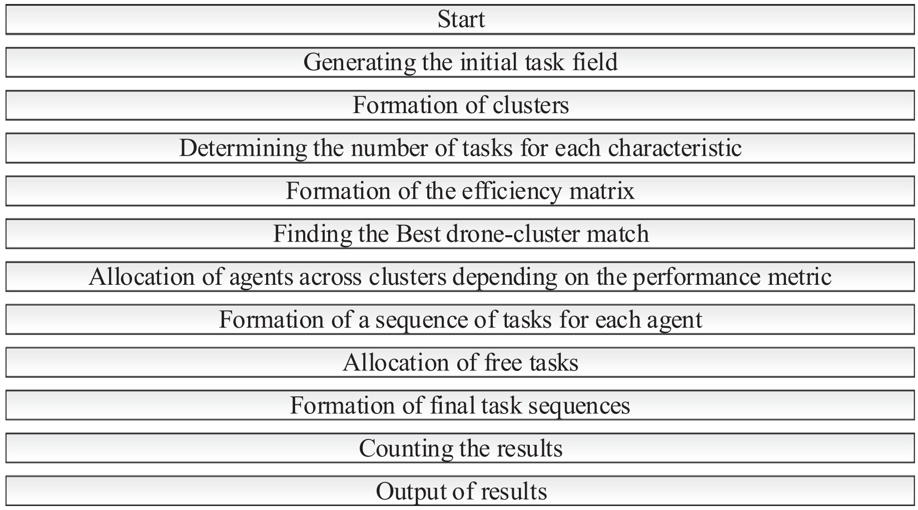

Fig. 3. Graphical description of the simulation program.

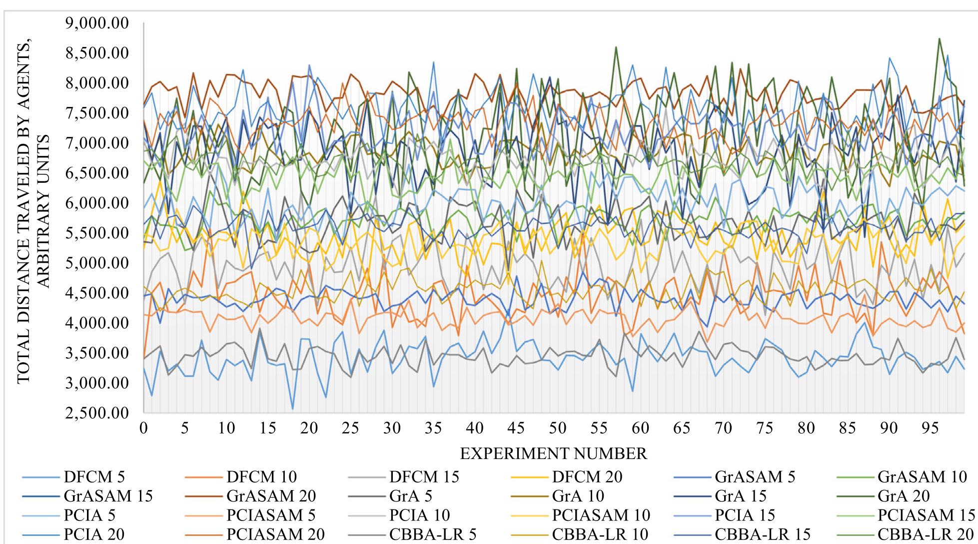

Fig. 4. Graphs of the total path traveled by agents when performing tasks.

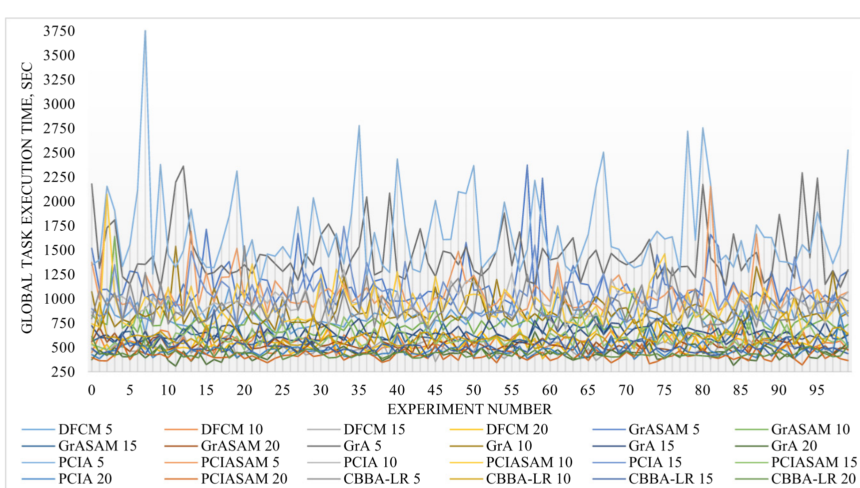

Fig. 5. Graphs of the time spent by agents when performing tasks.

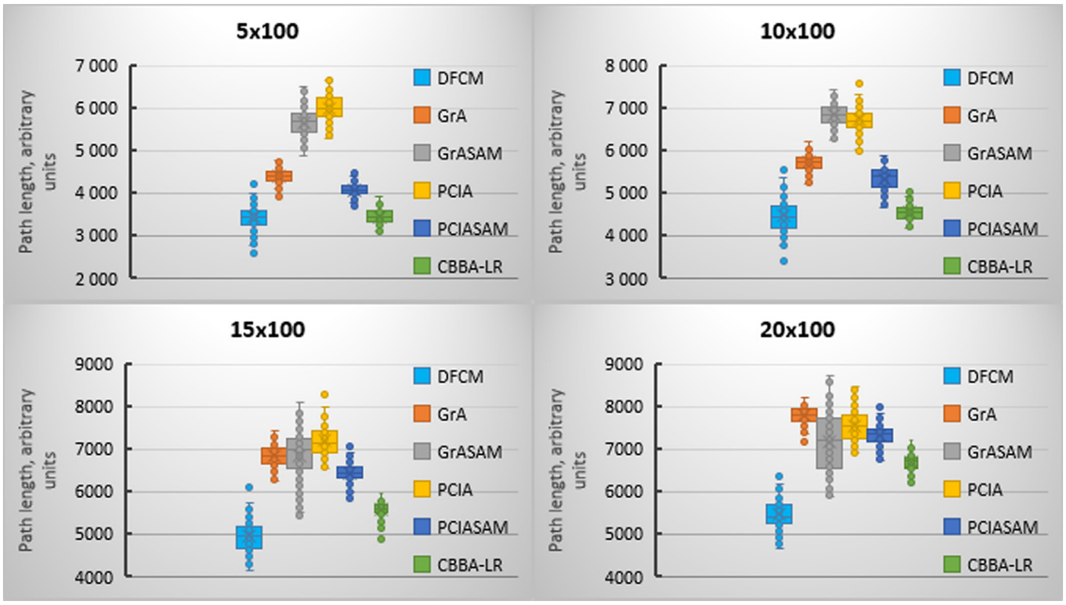

Fig. 6. The total distance traveled by 5, 10, 15 and 20 agents when performing 100 tasks.

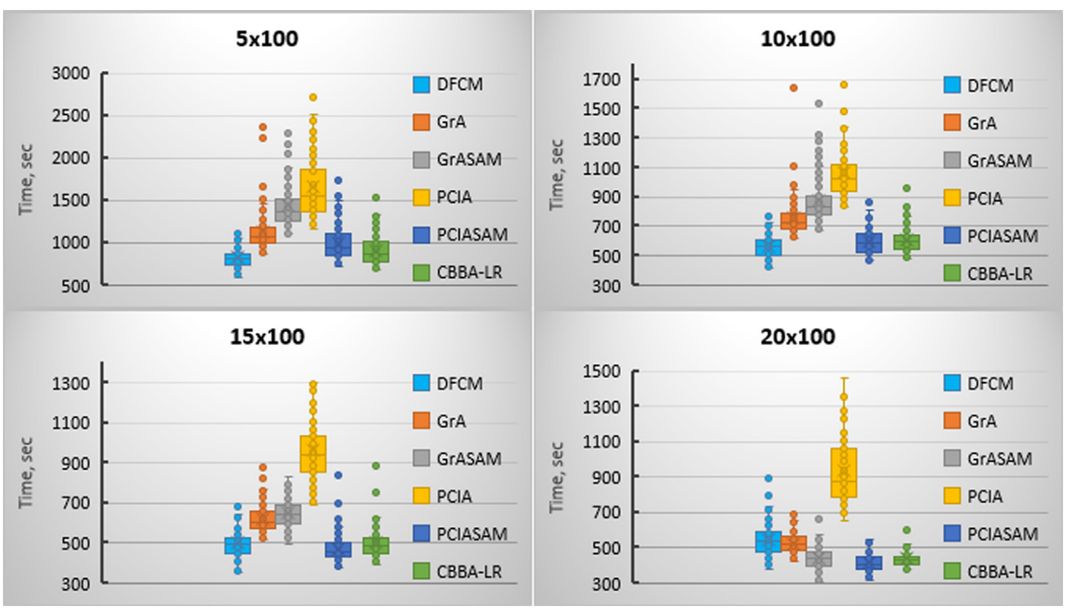

Fig. 7. The total time of performing 100 tasks by 5, 10, 15 and 20 agents when performing 100 tasks.

The analysis of the graphs presented in Fig. 6 allows concluding that the proposed method of task allocation with 15 and 20 agents per 100 tasks shows significantly good results of the metric of the path traveled compared to all the studied analogues. With 5–10 agents per 100 tasks, it shows significantly good results in comparison with methods of collective plan improvement and greedy algorithms, while the CBBA-LR algorithm is only slightly inferior to the proposed DFCM. Thus, it can be concluded that it is advisable to use DFCM to minimize the path traveled by agents and, as a result, minimize the energy spent on movement.

Analysis of the graphs presented in Fig. 7 allows concluding that the proposed method DFCM of task allocation with 5 and 10 agents per 100 tasks shows good results in terms of time spent in comparison with methods of collective plan improvement and greedy algorithms, while the CBBA-LR algorithm is only slightly inferior to the proposed DFCM. The effectiveness of the method changes significantly with an increase in the number of agents. The time spent performing tasks by the DFM algorithm with an increase in the number of agents makes it ineffective in comparison with analogues.

For a more accurate analysis, let's consider the results of the average distance traveled and the average task completion time for 5, 10, 15 and 20 agents when performing 100 tasks, which are presented in Tables 1 and 2, respectively. In the tables, the best values of time or distance for the method of task allocation in the UAV group are highlighted in bold. Fig. 8 and Fig. 9 show the results of the studies presented in histograms.

The results of experiments showed that the proposed method of task allocation in a clustered field of tasks reduces the average total distance traveled by agents when performing 100 tasks by 16%,

Table 1
The average distance traveled when the agents of the UAV group perform tasks in clusters.

S distance	DFCM	GrA	GrASAM	PCIA	PCIASAM	CBBA-LR
5x100	**3424,12**	4395,45	5670,36	5997,27	4077,16	3448,37
10x100	**4457,12**	5716,99	6844,80	6713,72	5336,42	4550,70
15x100	**4961,51**	6844,80	6859,71	7200,16	6443,82	5585,71
20x100	**5459,49**	7803,58	7160,26	7540,29	7322,94	6688,59

Table 2
Average time of the global task completion.

T TIME	DFCM	GrA	GrASAM	PCIA	PCIASAM	CBBA-LR
5x100	**823,80**	1130,96	1446,18	1669,65	997,36	907,89
10x100	**559,66**	752,90	873,78	1071,32	597,25	611,47
15x100	500,65	621,21	648,65	960,65	**473,10**	501,60
20x100	544,22	524,56	441,33	936,52	**415,34**	438,10

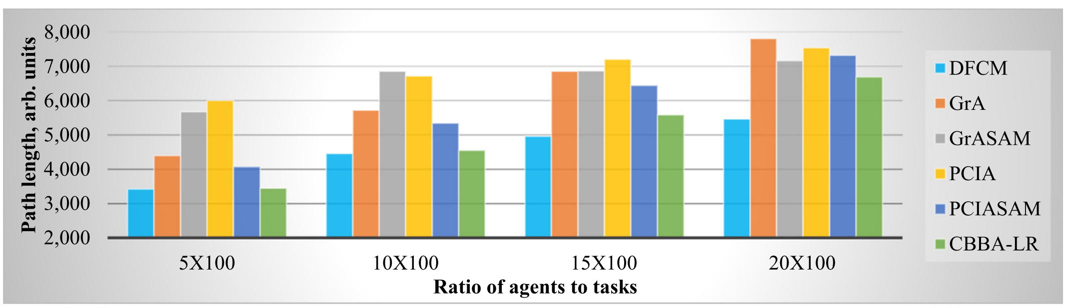

Fig. 8. Histogram of the average total path traveled by agents when performing tasks.

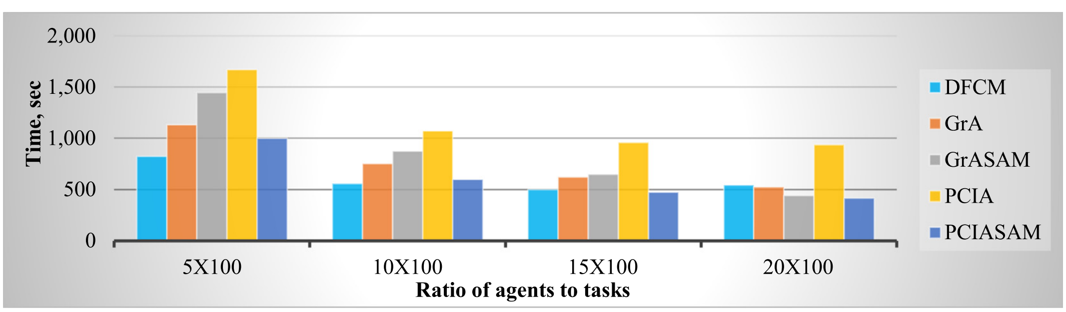

Fig. 9. Histogram of the average total time of the global task completion.

16.5%, 23% and 25.4% for 5, 10, 15, 20 agents, respectively, compared with the collective improvement of the plan algorithm with simulated annealing.

The proposed method of task allocation in a clustered field of tasks reduces the average distance traveled by agents when performing 100 tasks by an average of 20.1%, 22%, 27.5% and 30% for 5, 10, 15, 20 agents, respectively, compared with the greedy algorithm.

The proposed method of task allocation in a clustered field of tasks reduces the average distance traveled by agents when performing 100 tasks by an average of 39.6%, 34.8%, 27.6% and 23.7% for 5, 10, 15, 20 agents, respectively, compared with the greedy algorithm with simulated annealing.

The proposed method of task allocation in a clustered field of tasks makes it possible to reduce the average distance traveled by agents when performing 100 tasks by an average of 0.7%, 2.1%, 11.2% and 18.38% for 5, 10, 15, 20 agents, respectively compared to the consensus-based bundle algorithm with local replanning.

The proposed method of task allocation in a clustered field of tasks shows 17.4% and 6.3% reduction in task execution time for 5 and 10 agents compared to the algorithm of collective improvement of the plan with improved simulated annealing. At the same time, the results deteriorate for 15 and 20 agents and show that the PCIASAM algorithm is 5.8% and 31% more efficient for 15 and 20 agents, respectively. There is a negative trend of increasing the task execution time by the proposed method with an increase in the number of agents. Thus, the proposed method is effective in reducing the travel distance of agents, when the number of agents is relatively low compared to the number of tasks.

Can be noticed an interesting pattern. With an increase in the specific number of local tasks per agent (from 5 to 20 tasks per agent), the proposed method shows an increasingly worse dynamics of the results of the gap difference according to the metric of the path traveled, while under these conditions it is the best of the methods considered. However, under the same conditions, with an increase in the specific number of local tasks per agent (from 5 tasks per agent to 20 tasks per agent), the opposite shows the results of reducing the execution time of all tasks. So the result improves in direct proportion to the number of specific tasks per agent. Thus, with 20 tasks per 1 agent, the best result is obtained in comparison with analogues, provided that when the specific number of tasks was equal to 5, the method showed an ineffective result according to the metric of time spent.

Using these patterns and test results presented in Figs. 4-9, it can be determined that under certain conditions the proposed method is the most effective for all considered metrics under certain conditions, which is the main scientific result of this paper.

3.2. Evaluation of computational complexity

Let consider a comparison of computational complexity of the proposed method and the studied analogues.

The proposed method. The total number of tasks is N. The number of tasks in the i-th cluster, $i \in [1, M]$, is N_i, and the number of agents is M.

The set number of iterations of the simulated annealing method is v.

1. Allocation, formation of clusters of M pieces, matching task – cluster: $M * N$ actions.
2. Formation of the efficiency matrix for each cluster of agents. Each agent forms its own efficiency array of dimension M and sends data to the other agents. Total number of calculations is $M * N$ is the number of operations performed by all agents.
3. Calculation of the best agent–cluster combination. Agents divide all permutations for iteration into M intervals and choose a personal permutation corresponding to the agent index. At $M < 10$, each agent performs calculations: $\frac{M!}{M} = (M-1)!$. The total number of calculations is $M!$. At $M \geq 10$, each agent performs v calculations. The total number of calculations for all agents is $M * v$.

Forming the path of each agent based on combining the tasks of the agent cluster and tasks that are not included in the cluster. Sorting agents by distance traveled: $M * logM$. Task allocation to agents (taking into account the averaging criterion of the distance traveled), the number of remaining tasks is taken as N (estimate from the top): $N * M$. Total number of calculations: $M * (logM + N)$. The total number of calculations (top estimate, $M < 10$) will be:

$$V_{m<10} = MN + MN + M! + M(logM + N) = M(3N + (M-1)! + logM) \quad (17)$$

The total number of calculations (top estimate, $M \geq 10$) will be:

$$V_{m>10} = MN + MN + Mv + M(logM + N)) = M(3N + v + logM) \quad (18)$$

The algorithm of collective improvement of the plan with the improvement of simulated annealing. The designations of the number of tasks N and the number of agents M introduced.

The set number of iterations of the simulated annealing method: v.

1. Formation of the efficiency matrix for each cluster of agents. Each agent forms its own efficiency array of dimension N and sends data to the other agents. Total number of operations spread across all agents is $M * N$.
2. Calculation of the best agent-cluster combination. Agent a_i finds the maximum value of $\Delta d_i : N$. Number of iterative cycles equals N. Total number of calculations is $N * N$.
3. Comparison of the maximum element with elements of other agents: $M - 1$. The number of iterative cycles is N. Total number of calculations equals $(M-1)N$.

The total number of calculations (estimated from above) is:

$$V_{kp} = 2MN - N + N^2 \quad (19)$$

The algorithm of CBBA-LR. The designations of the number of tasks N and the number of agents M introduced.

1. The number of iterations of the method is equal to the number of tasks of the original field: N.
2. To compile a list of excluded points of each agent requires: N.
3. The formation of new sequences of each agent requires: M.

Table 3
Computational complexity of DFCM, PSIASAM and CBBA-LR for 20 agents and 100 tasks.

Number of tasks	Number of agents	DFCM	PCIASAM	CBBA-LR
100	1	301	10,100	10,100
100	2	603	10,300	10,200
100	3	907	10,500	10,300
100	4	1214	10,700	10,400
100	5	1536	10,900	10,500
100	6	1936	11,100	10,600
100	7	2840	11,300	10,700
100	8	7464	11,500	10,800
100	9	43,049	11,700	10,900
100	10	365,913	11,900	11,000
100	11	4337	12,100	11,100
100	12	4642	12,300	11,200
100	13	4947	12,500	11,300
100	14	5252	12,700	11,400
100	15	5557	12,900	11,500
100	16	5863	13,100	11,600
100	17	6168	13,300	11,700
100	18	6474	13,500	11,800
100	19	6779	13,700	11,900
100	20	7085	13,900	12,000

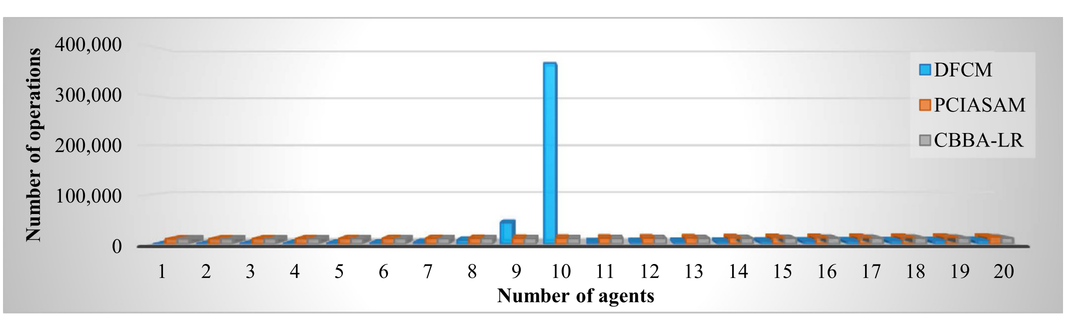

Fig. 10. Graph of changes in computational complexity with increasing number of agents.

The total number of calculations: $N(N+M) = NN + NM$.

Let's build a table of computational complexity of the proposed method, the method of collective plan improvement with improved simulated annealing and consensus-based bundle algorithm with local replanning, taking into account the increase of agents to 20 units for 100 tasks (Table 3).

The results shown on in Fig. 10.

Based on the analysis of computational complexity, it should be noted that the number of operations of the proposed method is growing rapidly with an increase in the number of agents from 7 to 10 units. 10 is a computationally valid number and the iteration method stops there, as can be seen from the graph. The restriction on the transition of calculation by simulated annealing allows reducing computational complexity. Thus, all three considered methods implemented on current UAV and MRS applications in general.

3.3. Evaluation of memory costs

The proposed method.

The designations of the number of tasks N and the number of agents M introduced. Storing information about tasks requires $4N$ bytes of memory. Storing cluster information requires $4M$ bytes of memory. Dynamic storage of efficiency matrix requires $4MM + 8M$ bytes of memory. Storing information about tasks of each agent requires $24M + N$ bytes of memory. The total value of the occupied memory: $4MM + 36M + 5N$.

The algorithm of collective improvement of the plan with the improvement of simulated annealing.

Table 4
Memory costs in bytes of DFCM, PCIASAM and CBBA-LR for 20 agents and 100 tasks.

Number of tasks	Number of agents	DFCM	PCIASAM	CBBA-LR
100	1	540	16,400	524
100	2	588	32,800	548
100	3	644	49,200	572
100	4	708	65,600	596
100	5	780	82,000	620
100	6	860	98,400	644
100	7	948	114,800	668
100	8	1044	131,200	692
100	9	1148	147,600	716
100	10	1260	164,000	740
100	11	1380	180,400	764
100	12	1508	196,800	788
100	13	1644	213,200	812
100	14	1788	229,600	836
100	15	1940	246,000	860
100	16	2100	262,400	884
100	17	2268	278,800	908
100	18	2444	295,200	932
100	19	2628	311,600	956
100	20	2820	328,000	980

The designations of the number of tasks N and the number of agents M introduced. Storing information about each task requires $4N$ bytes of memory. Dynamic storage of efficiency matrix requires $4NM + 8M$ bytes of memory. Storing information about tasks of each agent requires 24*M + N bytes of memory. The total value of the occupied memory:.$5N + 32M + 4NM$

The algorithm of CBBA-LR.

Storing information about each task requires $4N$ bytes of memory. Storing information about tasks of each agent requires $24M + N$ bytes of memory. The total value of the occupied memory: $24M + 5N$.

Let's build a table of memory costs of the proposed method, the method of collective plan improvement with simulated annealing and consensus-based bundle algorithm with local replanning, taking into account the increase of agents to 20 units per 100 tasks (Table 4).

The results shown on in Fig. 11.

Based on the estimate of memory costs, the following conclusion can be drawn: the proposed method can be implemented in the UAV group and the MPC as a whole, but it is worth noting that the PCIASAM algorithm shows a linear increase in memory costs as the number of agents in the group increases.

4. Discussion

The studied analogues of GrA, PCIA and their improvements showed significantly worse results in the parameter of the traveled path (from 16% to 33.8%), while with the number of agents from 10 units, the time to complete the global task by the method of collective plan improvement with the proposed improvement of PCIASAM annealing simulation is significantly better (up to 31% with 20 agents) of the proposed method. Perhaps these indicators depend on the clustering parameters and the inclusion of the simulated annealing method, which gives suboptimal solutions. In order to improve the performance characteristics of the method over time, in subsequent studies it is proposed to introduce an additional time cost optimization function along with the "leveling" function of the traveled path into the task allocation algorithm.

An interesting pattern observed in comparison with the CBBA-LR method. With the ratio of the number of agents to the number of tasks as 5:100, the values of the metrics of the traveled path are almost equal, but the execution time of the global DFCM task is 9.2% better than CBBA-LR.

At the same time, with a ratio of 20:100, the distance traveled by DFCM is 18.3% less, but the task execution time is 24.2% worse. The inverse proportionality of solutions is observed. It is worth noting that DFCM created specifically for conditions where the number of tasks exceeds the number of agents, while CBBA-LR is an algorithm for a dynamic environment.

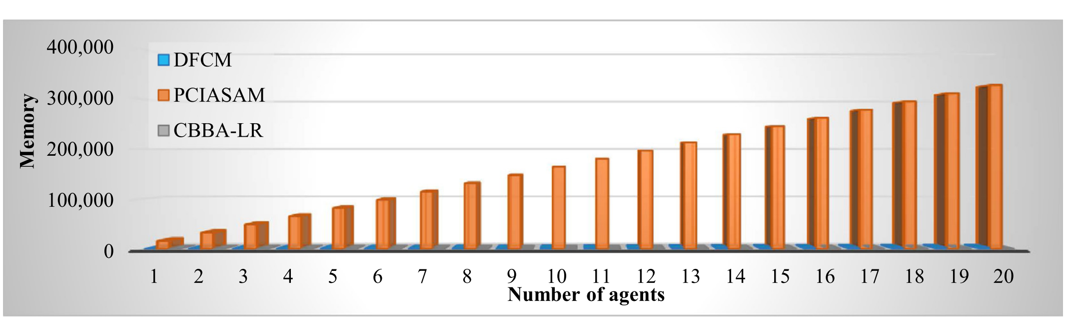

Fig. 11. Graph of changes in memory costs with increasing number of agents.

In this regard, in the future it is planned to conduct additional studies and modifications of the algorithm in the presence of several agent departure bases and dynamic tasks, both in time and in the space of the expandable field of tasks.

5. Conclusion

This article describes the development and evaluation of the effectiveness of the method of task allocation in a group of UAVs in a clustered field of heterogeneous tasks with a significant excess of the number of tasks over the number of agents by 5–20 times. The proposed method based on the allocation function of tasks in a clustered field of tasks according to the criterion of minimizing the path traveled by agents based on collective decision-making according to majority criteria.

To evaluate the effectiveness of the proposed method, it compared with the greedy task allocation algorithm, the algorithm for collective plan improvement and their improved versions with the construction of a flight task by simulated annealing and consensus-based bundle algorithm with local replanning. To conduct experimental research, a mathematical model implemented in the C++ environment. Based on the results of 2400 experiments, it can be concluded that the proposed method of task allocation is promising to reduce the path traveled by agents and, as a consequence, minimize the energy expended.

The proposed method makes it possible to reduce the distance traveled by a group of 5 to 20 agents when performing 100 tasks by an amount from 16% to 25.4% in comparison with the considered analogue of the method of collective plan improvement with modification of simulated annealing. At the same time, the task execution time by the proposed method with 15 and 20 agents per 100 tasks shows worse results by an amount from 17.4% to 31%. The improvement of the indicators of the traveled path is made by increasing the time to complete the global task. At the same time, for the number of agents from 5 to 10 per 100 tasks, the method shows consistently better performance of tasks in comparison with analogues, both in terms of the metric of the path traveled and in terms of task completion time.

In comparison with the consensus-based bundle algorithm with local replanning, a consistently good result observed with 5–15 agents and 100 tasks, however, with an increase in the number of agents to 20, a deterioration in the metric of time spent is observed. Based on the results of the study, it is noticeable that the proposed method has a wide potential for its application and development. The inverse proportionality of the results with the CBBA-LR method observed in wider ranges of the ratio of the number of tasks and the number of agents, which is interesting for future research.

The computational complexity of the method at the peak point with 10 agents is no more than $4 * 10^5$ operations, and the amount of memory consumed by the method allows it to be used on current UAV and MPC applications in general.

In the continuation of the research, the authors plan to further develop the proposed method for working in conditions of a non-deterministic terrain map, several bases of departure and landing of agents and dynamic tasks, both in time and in the space of an expandable field of tasks. Additionally, in subsequent studies, cases will be considered when one or several agents will be needed for one heterogeneous task.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work supported by a grant from the President of the Russian Federation for young scientists - candidates of science (No. MK-300.2022.4 Development of methods and algorithms for the UAV swarm control system when performing heterogeneous tasks).

References

Berman, S., Halasz, A., Hsieh, M.A., Kumar, V., 2009. Optimized stochastic policies for task allocation in swarms of robots. IEEE Trans. Rob. 25 (4), 927–937. https://doi.org/10.1109/TRO.2009.2024997.
Bertsekas, D., Castanon, D., 1991. Parallel synchronous and asynchronous implementations of the auction algorithm. Int. J. Parallel Comput. 17, 707–732.
Brutschy, A., Scheidler, A., Ferrante, E., Dorigo, M., Birattari, M., 2012. "Can ants inspire robots?" Self-organized decision making in robotic swarms. IEEE/RSJ Int. Conf. Intell. Robots Syst. 2012, 4272–4273. https://doi.org/10.1109/IROS.2012.6386273.
Buckman, N., Choi, H.L., How, J.P., 2019. Partial replanning for decentralized dynamic task allocation. AIAA Scitech. https://doi.org/10.2514/6.2019-0915.
Chen, J., Qing, X., Ye, F., et al., 2022. Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment. J. Supercomput. 78, 1712–1740. https://doi.org/10.1007/s11227-021-03940-z.
Chung, S., Paranjape, A.A., Dames, P., Shen, S., Kumar, V., 2018. A survey on aerial swarm robotics. IEEE Trans. Rob. 34 (4), 837–855. https://doi.org/10.1109/TRO.2018.2857475.
Cui, J.-H., Wei, R.-X., Liu, Z.-C., Zhou, K., 2018. UAV motion strategies in uncertain dynamic environments: a path planning method based on Q-learning strategy. Appl. Sci. 8, 2169. https://doi.org/10.3390/app8112169.
Husheng, W., Hao, L., Renbin, X., 2021. A blockchain bee colony double inhibition labor division algorithm for spatio-temporal coupling task with application to UAV swarm task allocation. J. Syst. Eng. Electron. 32 (5), 1180–1199. https://doi.org/10.23919/JSEE.2021.000101.
Kalyaev, I.A., Gaiduk, A.R., Kapustyan, S.G. Models and algorithms of collective control in groups of robots. – M.: FIZMATLIT, 2009. – 280 s.
Kong, X., Gao, Y., Wang, T., Liu, J. and Xu, W., 2019. Multi-robot task allocation strategy based on particle swarm optimization and greedy algorithm. In: 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 1643-1646, doi: https://doi.org/10.1109/ITAIC.2019.8785472.
Kowalczyk, W. Target Assignment Strategy for Scattered Robots Building Formation. In: Proc. of the 3rd Intern. Workshop on Robot Motion and Control. Poland, Poznan, 2002, pp. 181–185.
Liao, T., Socha, K., Montes de Oca, M.A., Stützle, T., Dorigo, M., 2014. Ant colony optimization for mixed-variable optimization problems. IEEE Trans. Evol. Comput. 18 (4), 503–518. https://doi.org/10.1109/TEVC.2013.2281531.
Liu, R., Seo, M., Yan, B., Tsourdos, A., 2020. Decentralized task allocation for multiple UAVs with task execution uncertainties. Int. Conf. Unmanned Aircraft Systems (ICUAS) 2020, 271–278. https://doi.org/10.1109/ICUAS48674.2020.9213989.
Liu, Y., Song, R., Bucknall, R., Zhang, X., 2019. Intelligent multi-task allocation and planning for multiple unmanned surface vehicles (USVs) using self-organising maps and fast marching method. Inf. Sci. 496, 180–197. https://doi.org/10.1016/j.ins.2019.05.029. ISSN 0020–0255.
Luo, L., Chakraborty, N., Sycara, K., 2015. Provably-Good distributed algorithm for constrained multi-robot task assignment for grouped tasks. IEEE Trans. Rob. 31 (1), 19–30. https://doi.org/10.1109/TRO.2014.2370831.
Majeed, A., Lee, S., 2018. A Fast Global Flight Path Planning Algorithm Based on Space Circumscription and Sparse Visibility Graph for Unmanned Aerial Vehicle. Electronics 7, 375. https://doi.org/10.3390/electronics7120375.
Mathew, N., Smith, S.L., Waslander, S.L., 2015. Planning paths for package delivery in heterogeneous multirobot teams. IEEE Trans. Autom. Sci. Eng. 12 (4), 1298–1308. https://doi.org/10.1109/TASE.2015.2461213.
Mouton, H., Roodt, J., Roux, H., 2011. Applying reinforcement learning to the weapon assignment problem in air defense. Scientia Militaria South African J. Military Stud. 39 (2), 1–15.
Msala, Y., Hamlich, M., Mouchtachi, A., 2019. A new robust heterogeneous multi-robot approach based on cloud for task allocation. In: 2019 5th International Conference on Optimization and Applications (ICOA), pp. 1-4, https://doi.org/10.1109/ICOA.2019.8727618.
Mukhedkar, R., Naik, S., 2013. Weapon target allocation problem using fuzzy model. Int. J. Application Innovation Eng. Manage. 2 (6), 279–289.
Murphy. R., 1999. Target-Based weapon target assignment problems. In: Nonlinear Assignment Problems: Algorithms and Applications. Kluwer Academic Publishers (Vol. 7), pp. 39–53.
Notomista, G., Mayya, S., Hutchinson, S., Egerstedt, M., 2019, An Optimal Task Allocation Strategy for Heterogeneous Multi-Robot Systems. In: 2019 18th European Control Conference (ECC), pp. 2071-2076. https://doi.org/10.23919/ECC.2019.8795895.
Oliveira, S., Hussin, M.S., Roli, A., Dorigo, M., Stützle, T., 2017. Analysis of the population-based ant colony optimization algorithm for the TSP and the QAP.

IEEE Congress on Evolutionary Computation (CEC) 2017, 1734–1741. https://doi.org/10.1109/CEC.2017.7969511.
Patel, R., Rudnick-Cohen, E., Azarm, S., Otte, M., Xu, H., Herrmann, J.W., 2020. Decentralized task allocation in multi-agent systems using a decentralized genetic algorithm. IEEE Int. Conf. Robot. Automat. (ICRA) 2020, 3770–3776. https://doi.org/10.1109/ICRA40945.2020.9197314.
Payton, D., Daily, M., Estowski, R., Howard, M., Lee, C., 2001. Pheromone robotics. Auton. Robot. 11 (3), 319–324.
Payton, D., Estkowski, R., Howard, M., 2005. Pheromone robotics and the logic of virtual pheromones. In: Proc. 1st Int. Workshop Swarm Robotics at SAB 2004, LNCS vol. 3342. Berlin, Germany: Springer-Verlag, pp. 45–57.
Petrenko, V.I., Tebueva, F.B., Ryabtsev, S.S., Gurchinsky, M.M., Struchkov, I.V., 2020. Consensus achievement method for a robotic swarm about the most frequently feature of an environment. IOP Conf. Ser.: Mater. Sci. Eng. 919, (4) 042025.
Petrenko, V., Tebueva, F., Ryabtsev, S., Sakolchik, A., Antonov, V., Makarenko, S., 2022. Iterative method of labor division for multi-robotic systems. Proc. Int. Conf. Artif. Life Robot., 699–702
Pshikhopov, V.K., Soloviev, V.V., Titov, A.E., Finaev, V.I., Shapovalov, I.O., 2015. Group control of moving objects in uncertain environments. In: Pshikhopova, V.Kh., Fizmatlit, M., p. 305.
Romeijn, H.E., Romero Morales, D., 2000. A class of greedy algorithms for the generalized assignment problem. Discret. Appl. Math. 103 (1–3), 209–235. https://doi.org/10.1016/S0166-218X(99)00224-3.
Shang, Y., 1844. Resilient multiscale coordination control against adversarial nodes. Energies 2018, 11. https://doi.org/10.3390/en11071844.
Shang, Y., 2021. Resilient consensus for robust multiplex networks with asymmetric confidence intervals. IEEE Trans. Network Sci. Eng. 8 (1), 65–74. https://doi.org/10.1109/TNSE.2020.3025621.
Shimaa, T., Rasmussena, S., Sparksa, A., Passino, K., 2006. Multiple task assignments for cooperating uninhabited aerial vehicles using genetic algorithms. Comput. Operations Res. 33, 3252–3269.
Sikanen, T., 2008. Solving Weapon target assignment problem with dynamic programming. Independent Res. Projects Appl. Math., 32 p
Soleimanpour-Moghadam, M. and Nezamabadi-Pour, H. 2020. Discrete Genetic Algorithm for Solving Task Allocation of Multi-robot Systems. In: 2020 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), pp. 006-009, doi: https://doi.org/10.1109/CSIEC49655.2020.9237316.
Task-Allocation // GitHub URL: https://github.com/BenJoice/Task-Allocation (date of the application: 02.02.2023).
Wei, T., Yongjiang, H., Yuefei, Z., Wenguang, L. and Xiaomeng, Z., 2021. Multi-UAV task allocation based on type mamdani fuzzy logic. In: 2021 7th International Symposium on Mechatronics and Industrial Informatics (ISMII), pp. 184-187. https://doi.org/10.1109/ISMII52409.2021.00046.
Wei, C., Ji, Z., Cai, B., 2020. Particle swarm optimization for cooperative multi-robot task allocation: a multi-objective approach. IEEE Rob. Autom. Lett. 5 (2), 2530–2537. https://doi.org/10.1109/LRA.2020.2972894.
Yu, J., LaValle, S.M., 2016. Optimal multirobot path planning on graphs: complete algorithms and effective heuristics. IEEE Trans. Rob. 32 (5), 1163–1177. https://doi.org/10.1109/TRO.2016.2593448.
Yuan, M., Ling, M.-X., Zeng, Q.-S., 2008. An AntColony algorithm based on pheromone declining for solving the WTA problem. Int. J. Computer Simulation. 25 (2), 23–25.
Zakiev, A., Tsoy, T., Magid, E., 2018. Swarm Robotics: Remarks on Terminology and Classification. In: Third International Conference, ICR 2018, Leipzig, Germany, September 18–22, 2018, Proceedings. https://doi.org/10.1007/978-3-319-99582-3_30.
Zavlanos, M., Spesivtsev, L., Pappas, G., 2008. A distributed auction algorithm for the assignment problem. In: Proc. of the IEEE Conf. on Decision and Control, pp. 1212–1217.
Zavlanos, M., Pappas, G., 2007. Sensor-based dynamic assignment in distributed motion planning. Proc. IEEE Intern. Conf. Robotics and Automation., 3333–3338
Zavlanos, M., Pappas, G., 2008. Dynamic assignment in distributed motion planning with local coordination. IEEE Trans. Rob. 24 (1), 232–242.
Zhang, Y., Feng, W., Shi, G., Jiang, F., Chowdhury, M., Ling, S.H., 2020. UAV swarm mission planning in dynamic environment using consensus-based bundle algorithm. Sensors 20, 2307. https://doi.org/10.3390/s20082307.
Zhang, J., Wang, X., Xu, C., 2012. ACGA algorithm of solving weapon target assignment problem. Open J. of Appl. Sci. 2 (4B), 74–77.
Zhao. H., Dorigo, M., Allwright, M., General Dynamic Neural Networks for the Adaptive Tuning of an Omni-Directional Drive System for Reactive Swarm Robotics. In: 2021 25th International Conference on Methods and Models in Automation and Robotics (MMAR), 2021, pp. 79-84, https://doi.org/10.1109/MMAR49549.2021.9528468.

Further reading

Nam, C., Shell, D.A., 2015. Assignment algorithms for modeling resource contention in multirobot task allocation. IEEE Trans. Autom. Sci. Eng. 12 (3), 889–900. https://doi.org/10.1109/TASE.2015.2415514.
Chopra, S., Notarstefano, G., Rice, M., Egerstedt, M., 2017. A distributed version of the hungarian method for multirobot assignment. IEEE Trans. Rob. 33 (4), 932–947. https://doi.org/10.1109/TRO.2017.2693377.