Labour division algorithm for a group of unmanned aerial vehicles in a clustered target field

*Fariza Tebueva, Vladimir Antonov, and Nikolay Svistunov**

Institute of Mathematics and Information Technologies named after Professor N.I. Chervyakov, North Caucasus Federal University, Stavropol, Russian Federation

Abstract. In this paper we propose an algorithm for tasks distribution (division of labour) for a group of unmanned aerial vehicles (UAVs) when monitoring an emergency zone. The input data of the algorithm are information on the homogeneous group of UAVs, the coordinates of the home point, and a set of elementary subtasks coming from the command center. The presented algorithm is analytical and allows obtaining the correct distribution result for any consistent input data. The algorithm is based on the principle of preliminary combining elementary tasks into clusters on a territorial basis. The results of simulation showed that the proposed labour distribution algorithm allows to achieve an average of 4.7% – 12.8% less time to complete a global task in comparison with the greedy algorithm. We experimentally established that the best result is achieved when choosing a cluster size so that about 75% of tasks are included in clusters, and 25% of tasks remain free.

1 Introduction

Modern emergency zones monitoring systems aim to counteract a negative tendency of changing the state of the environment, expressed in the activation of unfavourable, dangerous natural phenomena and processes and, accordingly, in an increase in the frequency and scale of natural disasters, the development of natural disasters into man-made and vice versa [1-2]. The growth of technological progress in the field of robotic, satellite and unmanned aircraft systems makes it possible to monitor the development of emergencies to assess the state of the environment, man-made objects, analyze the processes and phenomena occurring in them for the timely identification of trends in their change in order to ensure the prevention and elimination of natural and man-made emergencies [3-5].

The existing systems for monitoring emergency zones cannot be described as highly effective for various reasons. Classic aviation has limitations in use due to the high cost of aircraft, maintenance personnel, pilots and infrastructure. Due to the low resolution, satellite monitoring can leave unnoticed small emergency zones, which potentially pose a great danger, and updating information several times a day does not ensure high efficiency in eliminating the consequences of an emergency. Contact towers and sensor systems for

* Corresponding author: svistunovn4@gmail.com

monitoring potentially dangerous zones create difficulties in the transmission and processing of information. The lack of mobility does not allow tracking the dynamics of emergency zones. A need for constant use of human labour, limited area of monitoring and a high cost of construction and operation are significant disadvantages of these systems. It is worth noting that the listed monitoring tools are difficult to use, for example, for taking samples of the atmosphere or searching for people with a simultaneous monitoring of the emergency zone, which can be performed in parallel by unmanned aircraft. In light of the above circumstances, the most promising direction for solving the problems of automating the monitoring of emergency zones is the use of UAV groups.

With the help of UAV groups, a wide range of tasks, such as the tasks of monitoring the technical conditions and preventing the occurrence of emergencies (including the examination of the state of objects), as well as tasks related to the inspection of accident sites of significant dimensions, are solved [6]. At the same time, there are serious shortcomings in the autonomy of control systems for UAV groups, which are due to the complexity of objectively existing system connections, the patterns of interaction of group elements in conditions of environmental uncertainty and possible instability of elements. There are well-proven methods for controlling elements and small groups of UAVs, but in the absence of an autonomous system for decomposition and distribution of tasks within the elements of the group. The lack of a direct relationship with the systems for monitoring and forecasting the development of emergencies makes these systems of little use. The known approaches to modelling control systems for UAV groups when performing global tasks do not allow fully implementing autonomous systems for monitoring emergency zones.

The main research topics in the field of UAV groups control [7] are presented in Fig. 1.

Fig. 1. The main topics of research in the field of UAV groups control.

Many of today’s solutions for presented in Fig. 1 problems and tasks are only of theoretical importance for science, since there are still not enough practical examples of the use of UAV groups and swarms in the world. So in work [7] some promising properties of swarm robotics are highlighted. However, there are currently very few real-world commercial applications available. The reasons for this are very different. In the works of E. Sahin and A. Winfield [8], three of them are listed:

1. Algorithmization. Swarm robotics must design both physical robots and the behaviour of individual robots, so global collective behaviour arises from their interactions. Currently, there is no general method of transition from individual to group behaviour.

2. Implementation and testing. The use of many real robots requires a good laboratory infrastructure to be able to conduct experiments.

3. Analysis and modelling. Groups of robots are usually stochastic, nonlinear, so building mathematical models for testing and optimization is difficult. These models may be needed to build secure applications in the real world.

A swarm of robots can perform tasks in which the main goal is to cover a wide region. Robots can disperse and perform monitoring tasks, in particular, natural emergency areas. The main advantage is that the use of UAV groups allows you to quickly inspect large areas. At the same time, the swarm scalability makes it more reliable.

The main task in the modern use of group and swarm robotic systems is the need to distribute tasks between the agents of the system (individual UAVs). This task is widely described in the literature as the task of the division of labour. The division of labour problem can arise in multi-robotic systems and especially in swarm robotics. The organization of the work of the swarm can be built only by applying the methods of distribution of tasks. With a mixed architecture of management and decomposition of tasks on the server, it is possible to perform targeted operations in a swarm.

In [9-13], algorithms for the division of labour in multi-robotic systems are presented. In these works, for the distribution of tasks, such approaches are used as:

- keeping a history of actions, when each agent observes the other robots and independently performs empty tasks by adapting his behaviour to the current division of labour [10];
- the use of a scheme for transmitting information about agents, and interaction through light signals [11];
- selection of tasks in a random way, but with a constant time; brute force algorithm; an algorithm for sequential assignment of tasks using minimal communication, but a lot of time [12];
- the use of distributed algorithms based on the theory of automata, which solves the main problem by independently assigning a desired task to each robot [13].

2 Materials and methods

2.1 Analogue algorithm

As an analogue for the proposed algorithm, we chose a greedy algorithm, which is described in sufficient detail in [12-14]. The essence of the algorithm is the sequential execution of nearby tasks by agents. The value $r_{A_i}(m_j)$ is used as the metric of the preference of the task m_j by the agent $A_i \in A$, where A is the set of all agents. $r_{A_i}(m_j)$ estimates the time of agent A_i moving from its current coordinates (x_{A_i}, y_{A_i}) to the coordinates of the task $m_j (x_{m_j}, y_{m_j})$, $m_j \in M$, where M is the set of all tasks for the group:

$$r_{A_i}(m_j) = \frac{\sqrt{(x_{A_i} - x_{m_j})^2 + (y_{A_i} - y_{m_j})^2}}{s(A_i)}, \quad (1)$$

where $s(A_i)$ is the speed of agent A_i movement. At the beginning of the system operation, as well as after the execution of the current task, each agent $A_i \in A$ chooses a task to execute m_{A_i} based on the principle of minimizing the metric r_{A_i} :

$$m_{A_i} \leftarrow \arg \min_{m \in M} r_{A_i}(m). \quad (2)$$

Then the task is excluded from the list of tasks:

$$M \leftarrow M \setminus m_{A_i}. \quad (3)$$

There are many variations on this approach to task allocation. In particular, the coordinates of agents and tasks can be specified in three-dimensional space and the speed of movement of agents can differ in different directions. The metric $r_{A_i}(m_j)$ can also be defined in a different way.

The greedy algorithm is quite effective in situations where the list of tasks is not predetermined, but is replenished during the operation of the system. However, the use of a greedy algorithm when the list of tasks is known a priori often leads to insufficient planning efficiency. This problem becomes the most noticeable at the final stages of the global task performing. The last tasks remained unallocated can be at a sufficiently large distance from each other, from the current position of the agents and/or from the base point of the agents (Fig. 2).

Fig. 2. An example of a non-optimal task execution sequence generated by greedy algorithm.

2.2 Proposed algorithm

To overcome the shortcomings of the greedy algorithm, we propose a labour division algorithm, main idea of which is the preliminary partial or complete distribution of the existing tasks into clusters. The input data of the algorithm are:

- a number of agents n_A ;
- a set of tasks $M \neq \emptyset$, each task $m_j \in M$ is described by its coordinates (x_{m_j}, y_{m_j}) ;
- a number of tasks in the cluster n_c ;
- coordinates of the base point $B(x_B, y_B)$.

The algorithm consists of two stages: the formation of task clusters and the execution of tasks by agents.

First stage. Formation of task clusters.

1. Define initial set of empty clusters $C = \{C_i = \emptyset \mid i = \overline{1, n_A}\}$.
2. $i \leftarrow 0$.
3. While $i < n_A$ and $M \neq \emptyset$:
 - 3.1. Find in the task list M task m , which is the most distant from B .
 - 3.2. $C_i \leftarrow C_i \cup \{m\}$.
 - 3.3. $M \leftarrow M \setminus m$.
 - 3.3. While $|C_i| < n_c$ and $M \neq \emptyset$:
 - 3.3.1. Calculate the center coordinates $Z_i(x_{z_i}, y_{z_i})$ of the cluster C_i :

$$x_{z_i} \leftarrow \frac{1}{|C_i|} \sum_{m \in C_i} x_m; \quad (4)$$

$$y_{z_i} \leftarrow \frac{1}{|C_i|} \sum_{m \in C_i} y_m. \quad (5)$$

3.3.2. Find in the task list M task m' , which is the least distant from Z_i .

3.3.3. $C_i \leftarrow C_i \cup \{m'\}$.

3.3.4. $M \leftarrow M \setminus m'$.

3.4. $i \leftarrow i + 1$.

After the completion of the first stage, each agent A_i is assigned a cluster C_{A_i} , the expected time to move to the center of which is the smallest. In the case of using a homogeneous group of UAVs initially located at the same base point, the allocation is made by the number of agent.

At the second stage, each agent executes the tasks of its cluster using the greedy algorithm described in the previous section.

Second Stage. Task performance by agent $A_i \in A$.

1. If $C_{A_i} \neq \emptyset$:

1.1. Execute the tasks of the cluster C_{A_i} using the greedy algorithm.

Else:

1.2. If $(\bigcup_{i=1, n_A} C_i) \cup M \neq \emptyset$:

1.2.1. Select the nearest task $m' \in (\bigcup_{j=1, n_A} C_j) \cup M$ for execution.

1.2.2. Exclude m' from the set it belongs to (M or C_j).

Else:

1.2.3. Return to the home point.

Depending on the initial number of tasks in the set M , as well as the parameters n_A and n_c , the following situations may occur:

- all tasks are distributed across clusters (for $n_A \cdot n_c = |M|$);
- not all tasks are distributed across clusters (for $n_A \cdot n_c < |M|$);
- there are incomplete and/or empty clusters (for $n_A \cdot n_c > |M|$).

In any of these situations, the algorithm has convergence; however, in order to gain an advantage over the greedy algorithm, it is necessary to determine the principle of choosing the number of tasks in the cluster.

To determine the optimal values of the parameters of the proposed algorithm, as well as to assess its effectiveness in comparison with the greedy algorithm, a number of computational experiments were carried out. For each experiment, the size of the working field is 40x40 cells, the number of agents is 6, 10 or 14, the number of tasks is 60 or 80, the cluster size is being changed from 0 to 15 tasks with the step of 1, where “0” corresponds to the greedy algorithm. The speed of movement of agents is 1 unit of length, equal to the side of the cell, per unit of time. One task is performed by an agent in 10 units of time. Mission and home positions are generated using a uniformly distributed random numbers generator (RNG). The results of computational experiments are presented in the next section.

3 Results

Figure 3 shows the results of 1000 computational experiments for each of the configurations presented in the previous section. Comparison of the global task execution time when changing the cluster size within the same configuration was carried out on the same set of initial data. The time of the global task completion is understood as the time when the last agent returned to the base point.

Fig. 3. Global task accomplishment time in the experiment.

The experimental results confirm the possibility of increasing the efficiency of task distribution using the proposed algorithm. However, in order to obtain better results in comparison with the greedy algorithm, it is required to choose the optimal cluster size that allows achieving the smallest average time of the global task execution. According to the results of the experiment (Table 1), the smallest average execution time for a global task is achieved with such a choice of the cluster size so that about 25% of tasks remain unallocated.

Table 1. Optimal cluster size and corresponding percent of distributed tasks.

Number of tasks	Number of agents		
	6	10	14
60	8 (80%)	4 (67%)	3 (70%)

80	11 (83%)	6 (75%)	5 (70%)
----	----------	---------	---------

To confirm the independence of the optimal cluster size from the spatial distance of the tasks, an additional experiment was carried out for a field of 1000x1000 cells (Fig. 4).

Fig. 4. Dependence of the global task execution time on the cluster size for increased field size.

An increase in the field size leads to a change in time, but does not lead to a change in the principle of choosing the optimal cluster size. The best overall result in minimizing the average time, as well as reducing the standard deviation, is achieved with a cluster size of 6, i.e., when 60 tasks out of 80 are distributed (75%).

The average advantage of the proposed algorithm in terms of the global task execution time for cluster sizes from Table. 1 varies from 4.7% to 12.8% depending on the number of tasks and agents (Table 2).

Table 2. Reduction in the average task execution time when using the clustering algorithm.

Number of tasks	Number of agents		
	6	10	14
60	7.4%	10.8%	11.6%
80	4.7%	9.5%	12.8%

Thus, the proposed algorithm can be used to reduce the average tasks execution time, provided that the correct cluster size is chosen.

4 Discussion

The proposed algorithm for the division of labour is more efficient in comparison with the selected analogue, the greedy algorithm. Reducing the execution time of a global task is achieved by explicitly including the most distant tasks from the basing point into the cluster. This makes it possible to reduce the likelihood of a significant territorial fragmentation of elementary tasks at the final stages of a global task.

The proposed algorithm is initially focused on solving the problem of monitoring a natural emergency zone, namely a forest fire, the logic of which lends itself well to decomposition into elementary subtasks. The study of the applicability and adaptability of the proposed

algorithm for more complex situations is one of the further directions of the development of this topic. These areas may include:

- consideration of the factor of energy costs, as well as the need for periodic charging / refuelling of agents;
- modelling of situations of refusal of agents;
- taking into account environmental conditions (relief, wind, etc.);
- distribution of tasks in a heterogeneous group of UAVs, differing in the way of movement in space, speed of movement, speed of task execution, specialization;
- heterogeneity of elementary tasks by type and volume;
- distribution of tasks in changing conditions (dynamic inclusion / exclusion of agents and tasks, weather changes, etc.);
- introduction of several base stations, differing in capacity (the number of agents serviced at the same time) and specialization (stations only for agents of a certain type), as well as mobile base stations.

The proposed algorithm for the division of labour in the considered conditions always allows one to obtain a feasible solution, but does not guarantee its optimality. The construction of optimal chains of subproblems requires the solution of optimization problems with high computational complexity, which is why this approach is not considered in this work. For situations with a relatively small number of agents and tasks, it is necessary to develop an optimal solution by enumerating possible options (including those with preliminary cutting off of obviously non-optimal ones) and choosing the best one among them. Thus, one of the issues for further study is the assessment of the optimality of the proposed algorithm for the division of UAV labour.

As the basis of the proposed algorithm for the division of labour, a greedy algorithm is used, following which the agent chooses a task with the minimum expected time to move to it as a task for execution. However, the proposed clustering algorithm can be used as a superstructure over other basic algorithms, for example, a greedy algorithm that maximizes the expected efficiency of the task [15-16]. The study of these aspects of using the clustering algorithm is planned to be carried out in the course of further research.

5 Conclusion

We presented and explored an algorithm for tasks distribution for a group of UAVs. This algorithm is analytical and allows for any consistent input data to work out the correct distribution of problems.

Comparison of the proposed algorithm with the greedy algorithm in a two-dimensional simulation model showed that the proposed task distribution algorithm allows achieving, on average, 4.7% – 12.8% less execution time of a global task. In this case, the choice of the cluster size has a significant impact on the efficiency of the algorithm. It has been experimentally established that the best result is achieved when choosing the size of the cluster in such a way that about 75% of the tasks are included in the clusters, and 25% of the tasks remain free.

References

1. J. Wu, J.E.M. Baartman, J.P. Nunes, L. Degrad. Dev., **32**, 640 (2021)
2. J.D.D. Benavides-Solorio, L.H. MacDonald, Int. J. Wildl. Fire, **14**, 457 (2005)
3. A. Belbachir, J.A. Escareno, *ICINCO 2016 - Proc. 13th Int. Conf. Informatics Control. Autom. Robot.* (SciTePress, 2016)

4. V. Sherstjuk, M. Zharikova, I. Dorovskaja, *Proc. 2020 IEEE 3rd Int. Conf. Data Stream Min. Process. DSMP 2020* (Institute of Electrical and Electronics Engineers Inc., 2020)
5. A.C. Kapoutsis, I.T. Michailidis, Y. Boutalis, E.B. Kosmatopoulos, *Comput. Electr. Eng.*, **91** (2021)
6. M. Ayamga, S. Akaba, A.A. Nyaaba, *Technol. Forecast. Soc. Change*, **167** (2021)
7. S.-J. Chung, A.A. Paranjape, P. Dames, S. Shen, V. Kumar, *IEEE Trans. Robot.*, **34**, 837 (2018)
8. E. Şahin, A. Winfield, *Swarm Intell.*, **2**, 69 (2008)
9. V.I. Petrenko, F.B. Tebueva, V.O. Antonov, V.N. Smykova, V.E. Nechvoloda, *IOP Conf. Ser. Mater. Sci. Eng.* (Institute of Physics Publishing, 2020)
10. C. Jones, M.J. Matarić, *IEEE Int. Conf. Intell. Robot. Syst.* (2003)
11. F. Ducatelle, A. Förster, *9th IEEE/RAS ...* (2009)
12. J. McLurkin, D. Yamins, *Robot. Sci. Syst.* (MIT Press Journals, 2005)
13. R. Groß, S. Nouyan, M. Bonani, F. Mondada, M. Dorigo, *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* (2008)
14. S. Ben Amar, P. Zong, *Aircr. Eng. Aerosp. Technol.*, **91**, 1245 (2019)
15. V.I. Petrenko, F.B. Tebueva, V.O. Antonov, N.Y. Svistunov, *IOP Conf. Ser. Mater. Sci. Eng.*, **1069**, 012045 (2021)
16. A. Zakiev, T. Tsoy, E. Magid, *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* (Springer Verlag, 2018)