DOI: https://doi.org/10.15276/aait.08.2025.10
UDC: 004.89

Heterogeneous swarm complex multilevel reconfiguration based on Swarm Chemistry

Mykyta S. Kozlov[1)]
ORCID: https://orcid.org/0009-0007-0071-7748; nikita.kozlov@stud.onu.edu.ua
Yuliia O. Shvets[1)]
ORCID: https://orcid.org/0009-0008-1714-3127; yulia.shvets@stud.onu.edu.ua
[1)] Odesa I. I. Mechnikov National University, 2, Dvoryanskaya Str. Odesa, 65082, Ukraine

ABSTRACT

The need for the rapid and safe analysis of areas affected by emergencies is driving the search for innovative approaches in the field of autonomous systems. One such solution is the use of swarms of unmanned aerial vehicles to scan territories. This study presents a hybrid approach to the dynamic control of swarms of drones, combining self-organisation mechanisms based on 'swarm chemistry' with the optimisation of global parameters using an evolutionary algorithm. The proposed multi-level swarm reconfiguration module allows the behaviour of individual agents and the entire formation to adapt in response to environmental changes. A clustering module has been implemented to divide the scanning area into subregions, as well as a route planning system that considers the scanning width of each individual drone. To effectively distribute zones between agents, a proportional algorithm is employed that considers the functional capabilities of each device. Simulation results showed a reduction in mission execution time of more than twofold and an increase in the stability of swarm behaviour, even under conditions of heterogeneous composition and partial agent loss. This approach has significant application potential in emergency monitoring, victim search and damage assessment. This study formulates the architecture of a dynamic swarm reconfiguration system that can adapt to environmental changes in real time. This approach ensures the system's flexibility and stability when performing tasks in difficult conditions. Further development of the model involves integrating machine learning methods to enhance adaptability and expanding to three-dimensional space to improve the accuracy and realism of simulations.

Keywords: Information technology; swarm intelligence; genetic algorithm, multi-agent systems; scan optimisation; swarm chemistry.

For citation: **Kozlov M. S., Shvets Y. O. "Heterogeneous swarm complex multilevel reconfiguration based on Swarm Chemistry".** ***Applied Aspects of Information Technology*****. 2025; Vol. 8 No. 2: 147–161. DOI: https://doi.org/10.15276/aait.08.2025.10**

INTRODUCTION

In modern conditions, society is increasingly facing challenges arising from man-made, natural and anthropogenic emergencies (Fig. 1). Large-scale fires, earthquakes and the destruction of infrastructure resulting from military action or accidents at critical infrastructure facilities require a rapid response, which is primarily dependent on the prompt receipt of reliable information about the affected area.

However, human involvement in such operations often poses critical risks to life due to toxic emissions, radiation contamination, landslide threats or combat operations.

This creates a need for autonomous aerial reconnaissance technologies that can scan terrain with minimal risk to personnel [1]. One of the most promising areas is the use of swarm systems based on unmanned aerial vehicles (UAVs).

Multi-level drone complexes demonstrate a high level of adaptability, scalability, and fault tolerance due to decentralised control, self-organisation, and flexible reconfiguration capabilities [2].

A literature review published in International Transactions in Operational Research, among numerous other studies, highlights the key role of drones in emergency response. These studies emphasise the ability of drones to quickly cover large areas, thereby minimising risks to rescuers and improving the accuracy of real-time data collection. This reaffirms the need for effective algorithms to coordinate and reconfigure swarms when scanning complex environments. The swarm approach is now widely recognised in scientific literature as the architectural foundation for emergency response systems, particularly for scanning large open areas and rapidly detecting objects of interest. This creates a need for autonomous aerial reconnaissance technologies capable of scanning terrain with minimal risk to personnel. One of the most promising areas is the use of swarm systems based on UAVs. Thanks to decentralised control, self-organisation, and the ability to reconfigure flexibly,

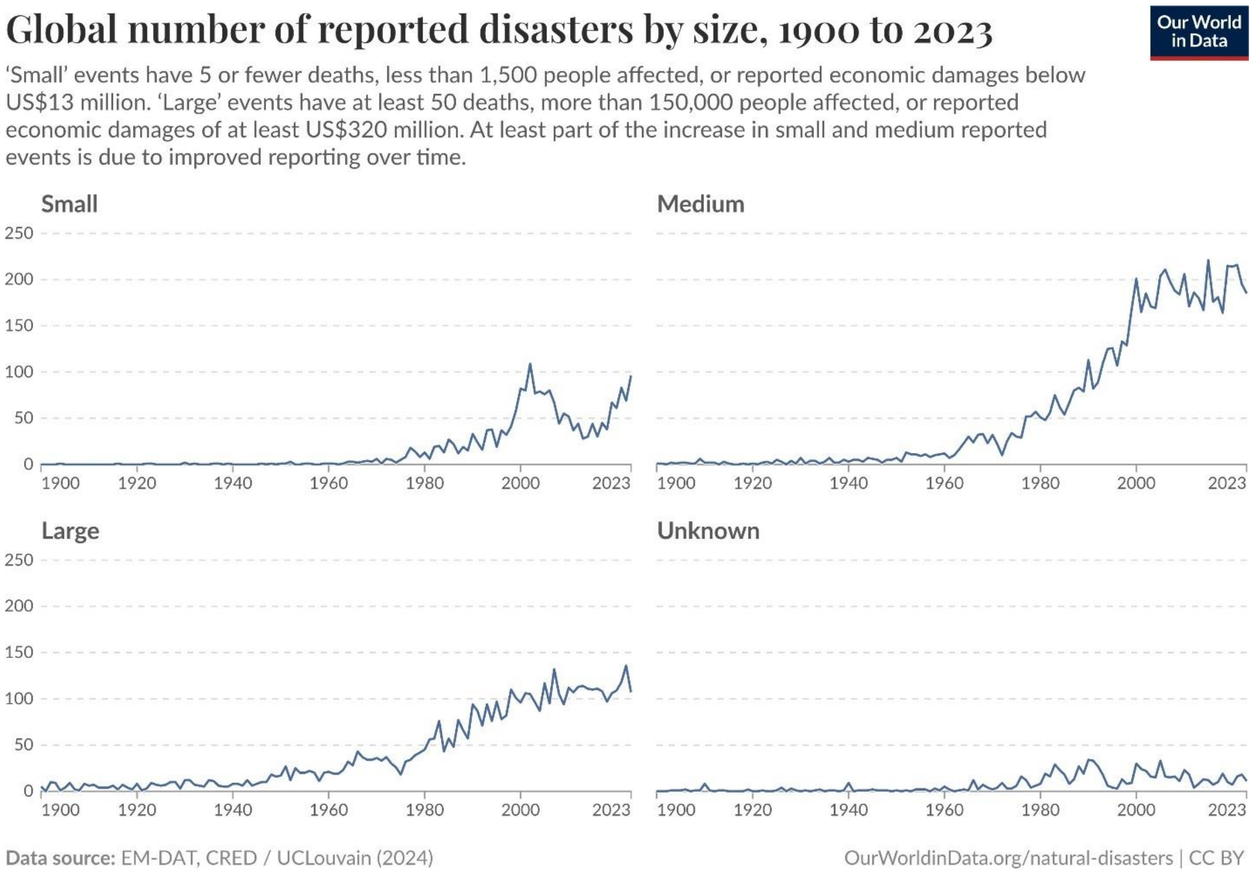

Fig. 1. **Global number of reported disasters by size, 1900 to 2023**

***Source*: compiled by the https://ourworldindata.org/grapher/natural-disasters-by-yearly-impact**

drone swarms demonstrate a high level of adaptability, scalability, and fault tolerance [3].

At the same time, the effectiveness of such systems depends largely on the configuration of the swarm – that is to say, the spatial arrangement of the drones within the scanning area.

This determines:

– the degree of coverage density;
– load uniformity;
– the speed at which the mission is executed;
– the extent of overlap or gaps.

Incorrect configuration can lead to duplication of efforts, reduced observation quality, increased mission time and energy overconsumption. Dynamic reconfiguration, which involves changing the formation during task execution when the swarm is heterogeneous, is particularly challenging. In such conditions, traditional centralised or rule-based approaches lack the necessary flexibility. Some studies, in particular [4], propose combined architectures using automata for dynamic swarm configuration aimed at achieving maximum coverage. However, these approaches mostly focus on modelling general movement structures, leaving the task of optimising reconfiguration to take individual drone parameters (e.g. scanning width) into account unresolved.

Modern approaches to swarm reconfiguration include:

– rule-based systems with fixed behaviour rules [5];
– global optimisation algorithms like particle swarm optimization (PSO) [6], ant colony optimisation (ACO) [7];
– hybrid schemes incorporating reinforcement learning components [7], [9].

However, without the necessary improvements, these methods have the following limitations:

– poor adaptation to real-time environmental changes;
– excessive computational complexity;
– predominance of static planning strategies;
– inefficiency in the event of partial agent loss.

The problem is further complicated by the fact that most swarm architectures are oriented towards a single control point or a single-level interaction model. However, recent studies have proposed multi-level architectures that allow planning, analysis and reconfiguration functions to be distributed between different components (e. g.

ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

swarm nodes, operators and wombs). Such a model is more relevant for multifunctional systems [10]. This work focuses on spatially adaptive reconfiguration of a swarm to efficiently scan a given territory, without complicating recognition roles or high-level strategies.

Multi-level models also introduce greater system resilience, as decision-making no longer relies on a single point of failure. In dynamic or adversarial environments, this decentralisation can be critical to maintaining operational continuity.

It should also be noted that existing solutions have not sufficiently addressed the issue of conflicts between drone trajectories, particularly when the same routes or parts of routes are traversed multiple times [11]. Works modelling a similar problem in graph form propose methods for sequentially modifying the search space by blocking arcs or vertices occupied by agents that have already laid out a route. One such approach is based on the sequential application of Dijkstra’s algorithm with dynamic graph restructuring to avoid conflicts between agents and has the potential to be applied to multi-drone scanning systems [12]. The problem is further complicated by the fact that most swarm architectures are oriented towards a single control point or single-level interaction model. Recent studies propose multi-level architectures that distribute planning, analysis, and reconfiguration functions between different components (e.g. swarm nodes, operators, and wombs). Such a model is more relevant for multifunctional systems [2], [10]. This work focuses on spatially adapting the reconfiguration of a swarm to efficiently scan a given territory without complicating recognition roles or high-level strategies.

Another important aspect of swarm reconfiguration is context-awareness – the ability of the system to interpret local environmental data (such as terrain obstacles, weather conditions, or dynamic no-fly zones) and use this information to adjust swarm structure or behaviour. Integrating onboard sensors and real-time communication allows drones to locally assess the feasibility of a route or scanning area, improving both safety and effectiveness. For example, in mountainous or forested regions, maintaining a uniform scanning pattern without context adaptation can lead to inefficient coverage or potential collisions with obstacles.

In addition, recent research has begun exploring the use of heterogeneous swarm configurations, where drones with different capabilities (e.g., flight time, sensor resolution, and communication range) cooperate in the same mission. In such scenarios, assigning roles and positions dynamically based on drone capabilities becomes essential for system performance. Heterogeneity not only increases task flexibility but also introduces additional complexity in planning, as configurations must consider load balancing, communication integrity, and role-dependent constraints. Solutions that can incorporate these variations in real time offer a significant advantage over static or homogeneous models.

Therefore, it is important to create an effective and scalable reconfiguration algorithm to ensure the following:

– coverage of any territory with a configurable coverage percentage;
– minimisation of overlap;
– reduction of mission time;
– adaptation to configuration changes in real time.

ANALYSIS OF LITERARY DATA

Existing approaches to reconfiguring swarm systems can be categorized based on their flexibility, adaptability, and complexity.

Currently, the most common approaches are based on:

– predefined rules (rule-based);
– swarm optimization algorithms (PSO, ACO);
– a combination of optimization algorithms and reinforcement learning (RL) methods.

Rule-based systems are widely used for tasks in relatively stable environments where scenarios are predictable. These systems operate using 'if-then' logic, ensuring low computational costs and predictable drone behaviour. They are effective for routine missions such as monitoring known objects or mapping flat terrain [13].

However, their main disadvantage is their inability to adapt to dynamic conditions. If environmental conditions change suddenly (e.g. the appearance of obstacles or the unpredictable geometry of the area), these systems are unable to adjust the swarm's strategy or configuration. Consequently, areas with duplicate scanning, ‘blind spots’, or inefficient drone usage arise.

A modern example of a rule-based system for managing a swarm of UAVs is the Multi-Objective O-Flocking (MO O-Flocking) model, which was developed by [14]. This model combines classic flocking architecture with physical-virtual laws and multi-criteria rule tuning based on adaptive parameters.

The MO O-Flocking model has a four-level architecture:

1) the sensor layer receives data from the drones' built-in sensors;

2) the decision layer applies rule-based logic to the current input data;

3) the action layer updates the flight speed and direction according to the calculated vectors;

4) evolutionary layer uses the ISPEA2 (Improved Strength Pareto Evolutionary Algorithm 2) algorithm to optimise rule parameters.

The behaviour of each agent is described by a set of forces influencing its speed change: repulsion, alignment, attraction, target orientation and obstacle avoidance.

This model is highly flexible thanks to its extensive range of configuration parameters. It also enables you to customise the behaviour of the swarm by taking target orientation, obstacle avoidance and swarm cohesion into account.

However, MO O-Flocking models have the following disadvantages:

– high configuration complexity due to the presence of 20 parameters, requiring significant computing resources;

– interdependence of parameters, whereby changing one parameter can significantly affect the behaviour of the entire system, complicating the interpretation of results;

– homogeneity of agents, which limits the model's suitability for tasks involving variable drone functionality.

One partial solution to the shortcomings of rule-based approaches, such as MO O-Flocking, is to use hybrid methods that combine PSO with reinforcement learning (RL) [15], [16]. One such approach is the PSO-M3DDPG algorithm, which was proposed in 2024 for pursuit and evasion tasks in multi-agent UAVs systems [17].

The PSO-M3DDPG algorithm combines the global search capabilities of PSO with the local optimisation capabilities of the Mini-Max Multi-Agent Deep Deterministic Policy Gradient (M3DDPG) algorithm. The aim of this approach is to improve the navigation of a swarm of UAVs in pursuit of evasive targets by learning effective agent strategies.

The PSO-M3DDPG algorithm combines the global search capabilities of the PSO algorithm with the local optimisation capabilities of the M3DDPG algorithm. This approach aims to improve the navigation of a swarm of UAVs in evasive target pursuit by learning effective agent strategies.

Unlike traditional rule-based methods, this hybrid model does not rely on pre-defined behaviours, allowing agents to autonomously adapt to complex, dynamic scenarios. This adaptability is especially crucial in real-world applications, where unexpected environmental changes can occur.

The PSO-M3DDPG architecture comprises the following steps [18]:

1) *initialization* – a population of policies (i.e. neural networks) representing different agent strategies is created;

2) *evaluation* – each policy interacts with the environment and its effectiveness is evaluated based on the accumulated reward;

3) *selection and mutation* – the best policies are selected as the ‘elite’, while the others undergo mutations to preserve diversity;

4) *learning* – M3DDPG uses the collected data to update the policy parameters and improve the agents’ strategies;

5) *population update* – the updated policy parameters are copied back to the population for subsequent iterations.

This model can adapt more easily to changes in the environment and is more effective with large numbers of agents in a swarm. However, its main drawback is its high computational complexity. Nevertheless, this method represents a promising direction for future research in autonomous swarm control.

PROBLEM STATEMENT

Currently, there is no universal approach that enables the flexible reconfiguration of the swarm according to local mission conditions, terrain features, threat dynamics and the characteristics of each drone.

This limitation becomes especially critical in time-sensitive or high-risk missions, where delays or inefficient coordination can lead to mission failure or increased vulnerability. Furthermore, without adaptive configuration, the swarm cannot fully leverage the heterogeneous capabilities of individual drones.

Solving this problem requires the development of technology that can rapidly generate and verify potentially effective configurations, thereby reducing the overall time taken and the duplication of scanning areas (Fig. 2).

The method being developed should take the following into account:

1. *The heterogeneity of the swarm.* Drones in a swarm may differ in:

– scanning area width (viewing angle, sensor range);

– movement speed.

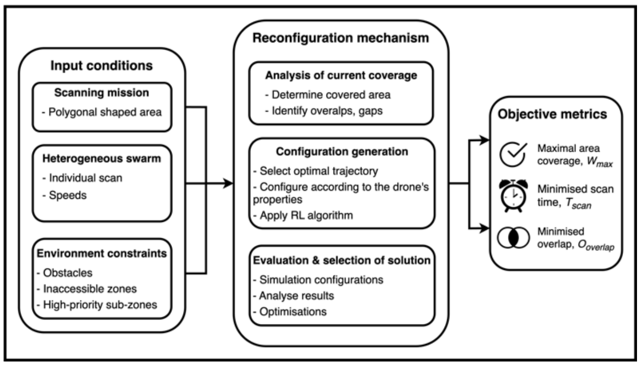

Fig. 2. **Scheme of reconfiguration of UAV swarms for efficient scanning**

***Source*: compiled by the authors**

This means that the reconfiguration approach must adapt to the capabilities of individual drones, ensuring differentiated route planning.

2. *Individual scanning widths.* Each drone has its own field of view, which may be determined by its design or depend on its flight altitude. This imposes additional requirements on trajectory generation:

– overlaps between fields of view must be avoided;

– complete coverage without gaps must be maintained;

– take width into account as a variable when optimizing coverage.

3. *Environmental dynamics.* Even if trajectories are planned at the start of the mission, the following may occur during scanning:

– new obstacles (e.g., buildings or weather phenomena);

– changes in the accessibility of individual areas;

– failure of one or more drones.

Therefore, the technology must support dynamic reconfiguration, i.e. the ability to change the spatial position and routes of drones based on new data in real time.

4. *Reconfiguration efficiency*. In this study, efficiency refers to specific formalized metrics that must be minimized or balanced:

– scanning time: the total time required to achieve the target percentage of area coverage;

– number of overlaps: the area scanned by two or more drones, indicating duplication of effort and loss of efficiency.

5. *Operational reconfiguration.* The system must support:

– local reconfiguration, for example, if an obstacle is detected by one drone;

– global reconfiguration, when the overall task topology changes (e.g., after the surveillance area is expanded).

This requires:

– coordination of actions at swarm level using a decentralized approach;

– use of information about current coverage and remaining unfilled areas.

RESEARCH GOAL

The study aims to improve the efficiency of swarm systems by developing adaptive information technology to reconfigure drone swarms using artificial intelligence algorithms. The proposed approach focuses on maximizing the scanning zone coverage area within the shortest possible time, while considering the swarm's heterogeneity, the width of individual scans and environmental dynamics.

To achieve this, the following tasks must be completed:

– analyse the subject area with a focus on hybrid approaches to complex swarm reconfiguration;

– explore the principles of swarm chemistry;

– investigate the application of genetic algorithms in swarm reconfiguration;

– define and set up the scan area;

– optimise the sequence of scanning subregions to ensure efficient and adaptive swarm operation.

The result of the research will be a complete system that:

– adapts to changes in the environment and the characteristics of the drones;

– minimises duplication and coverage time;

– works efficiently when scaling the swarm;

– allows flexible configuration changes in real time without loss of performance.

SWARM COMPLEX RECONFIGURATION HYBRID ARCHITECTURE

The heterogeneity of the swarm complex is addressed through a multi-level approach, enabling effective coordination of drones with diverse characteristics and functionalities. Rather than being treated as identical units, each drone is considered a specialised agent equipped with distinct capabilities – such as different flight endurance, sensor configurations or communication ranges.

Moreover, the multi-level architecture facilitates adaptive decision-making across various operational layers – from low-level trajectory planning to high-level strategic coordination – allowing the swarm to respond effectively to changes in the environment or to the loss of individual agents. At the level of individual nodes, swarm chemistry governs behaviour through the use of recipes – structured sets of behavioural rules that determine how each drone reacts to local stimuli and interactions with neighbouring agents. These recipes serve as the foundation for emergent group behaviour, enabling decentralised control and scalable coordination within the swarm.

The heterogeneity of the swarm complex is considered within the multi-level approach, which allows for effective coordination of drones with different characteristics and functionalities:

– at the level of individual drones (nodes), swarm chemistry determines the behaviour of each drone through recipes;

– at the swarm level, a genetic algorithm (GA) optimises global scan speed and width parameters based on individual recipes.

This approach enables flexible management of swarm structures and adaptation to a heterogeneous environment. Genetic algorithm allows optimising the initial parameters and structure of a swarm using evolutionary principles of selection and combinations of characteristics.

SWARM CHEMISTRY: CONCEPT AND PARAMETRIZATION FOR SWARM RECONFIGURATION

Swarm chemistry [19], [20], [21] is an extended version of the Boids algorithm that models drone interaction based on the following rules:

1) separation – avoids crowding of drones in a local group;

2) alignment – sligns the direction of movement with the average velocity vector of neighbours;

3) cohesion – striving for the average position of the group.

Swarm chemistry, unlike the classic Boids, formalises these rules in the form of a recipe or a set of kinetic rules, which allows us to adapt the behaviour of different types of drones.

Additionally, swarm chemistry defines the following kinetic principles:

1) straying – if there are no other agents within the local perception, random movement is performed;

2) randomness – means random control with a certain probability;

3) self-propulsion – adjusting the speed to the drone’s normal speed.

The recipe defines a set of kinetic parameters (KP) (Table 1) that characterise the behaviour of a particular type of drone. Unlike the classical approach, the recipe includes a scan width parameter that ensures adaptation to the environment, considering swarm heterogeneity. It is worth noting that the recipe specifies not only the speed and endurance of drones, but also the ability to cover the territory.

Table 1. **Kinetic parameters in recipes for configuring swarm behaviour**

KP	Name	Definition
KP1	R^i	Radius of local perception range
KP2	v_n	Normal speed
KP3	$v_{\min}$	Minimum speed
KP4	$v_{\max}$	Maximum speed
KP5	c_1	Strength of cohesive force
KP6	c_2	Strength of aligning force
KP7	c_3	Strength of separating force
KP8	c_4	Probability of random steering
KP9	c_5	Tendency of self-propulsion
KP10	s_1	Scanning width

***Source*: compiled by the authors**

Each recipe describes one type of drone, which means that a swarm can be either heterogeneous or homogeneous depending on the number of recipes that have been specified. If there is only one recipe, the swarm is homogeneous, but if there are two or more recipes, the swarm is heterogeneous.

GENETIC ALGORITHM

The genetic algorithm in the presented system is used for global optimisation of drone swarm parameters. The main goal of the GA is to determine the optimal combinations of speed, scanning width and other parameters that ensure efficient coverage of the territory and minimise energy consumption.

1. *Initialisation of the population*. Each individual in the population represents a set of parameters consisting of a combination of kinetic parameters from Table 1.

2. *Fitness assessment*. The fitness function is calculated based on the parameters of the scanning

route, the number of overlapping zones, energy consumption, and mission duration. In particular, the fitness function has the following form:

$$F = \alpha \cdot C + \beta \cdot E + \gamma \cdot T,$$

where *F* is overall quality assessment; *C* is territory coverage rate; *E* is energy consumption; *T* is mission time; α, β, γ are weighting factors.

3. *Selection*. A tournament selection is used, in which the best individual is selected from a group of individuals based on fitness. The size of the tournament determines the probability of selecting the best individual.

4. *Crossover*. The parameters of the two parents are combined to create offspring. A single-point crossover with a random split point is selected. If the dividing point is between the speed and scan width parameters, the new individual can get the speed from one parent and the scan width from the other.

5. *Mutation*. To avoid a local optimum, individual parameters are mutated with a certain probability. For example, the scan speed or width is changed to a random value within the permissible range specified by the recipe.

6. *Population update*. The best individuals from the current population are passed on to the next generation unchanged (elite selection), and the rest are replaced by new offspring.

7. *The stopping criterion*. The algorithm terminates when the specified number of iterations is reached, or the best solution is stable for a certain number of generations.

SETTING UP THE SCAN AREA

In order for the swarm to start scanning the area, we need to set the scan zone – the area $P \in R^2$ to be covered. This zone is described as a polygon of any shape and size (for example, a polygon with holes or irregular contours).

Here, R^2 denotes the two-dimensional Euclidean space, which means that the polygon *P* is defined by a set of points with real-valued coordinates (x, y), where $x, y \in R$. This allows for an accurate representation of the scan area in a continuous spatial domain.

Such flexibility in defining the area allows the system to be used in diverse environments, from urban landscapes with infrastructure obstacles to natural terrains like forests or coastlines. Accurately modelling complex scan zones is crucial for avoiding blind spots and ensuring full area coverage.

To ensure efficient and consistent coverage, the region *P* is divided into smaller subregions using KMeans [22] clustering and convex hull construction. This process consists of several sequential steps, which are described below.

The first stage is the generation of internal points. A regular grid *G* with a step size of *s* is constructed on the entire plane covering the polygon *P*, which can be represented by the following formula:

$$G = \{(x_i, y_j) |\ x_i = x_{min} + i \cdot s,\ y_j = y_{min} + j \cdot s\},$$

where x_i is coordinate along the horizontal axis (X); y_j is coordinate along the vertical axis (Y); $x_{\min}$, $y_{\min}$ are the minimum X and Y coordinates, respectively (i.e. the lower left point of the rectangle enclosing the polygon); *s* is grid spacing (distance between adjacent points in X and Y).

The indices i and j represent integer steps along the horizontal and vertical axes, respectively, and are used to systematically generate points across the entire 2D grid. Each point (x_i, y_j) corresponds to the intersection of the *i*-th vertical and *j*-th horizontal line in the grid.

Only the points inside the original polygon are selected from this set:

$$Points_{in} = \{(x, y) \in G |\ (x, y) \in P\},$$

where $Points_{in}$ is the subset of grid points that lie within the boundaries of the polygon *P*; $(x, y) \in P$ denotes that the point lies inside the polygon.

Based on the obtained internal points, the KMeans method is used to cluster the data into a given number of subregions *K* equals to number of drones. Such a specific number of clusters is chosen only during the initial model creation iteration.

This method divides the set of points into *K* clusters $C_1, C_2, C_3, ..., C_n$, so as to reduce the total square distance to the centres:

$$\min_{\mu_1,\ldots,\mu_K} \sum\nolimits_{i=1}^{K} \sum\nolimits_{x \in C_i} | x - \mu_i |^2,$$

where μ_i is the center of mass of the i-th cluster; x is cluster point.

The next step is that each cluster is transformed into a convex polygon R_i using the calculation of the convex hull:

$$R_i = \text{ConvexHull}(C_i) = \{y \in R^2 \mid y = \sum\nolimits_{k=1}^{n} \lambda_k x_k,\ \lambda_k \geq 0, \sum\nolimits_{k=1}^{n} \lambda_k = 1\Big\},$$

where: λ_k is non-negative weights (convex combination coefficients), where the sum of all $\lambda_k = 1$; y is any point inside the convex polygon defined by the points of the cluster C_i with all possible weights λ_k.

The resulting polygon is the smallest convex polygon containing all the points of the cluster C_i. The resulting envelope is then intersected with the original zone to avoid overlap:

$$R_i := R_i \cap P$$

As a result, a set of subregions $\{R_1, R_2, \ldots, R_K\}$ is obtained, which collectively cover the region *P*.

OPTIMISE THE SEQUENCE OF SCANNING SUBREGIONS

After the main scan area is divided into smaller subregions $\{R_1, R_2, \ldots, R_K\}$, the optimal order of their visits by the drone swarm is determined. This minimises the distance and, consequently, the time required for transitions between subregions, which directly affects the efficiency of the task. To do this, a greedy algorithm is used that approximates the traveling salesman problem (TSP).

Each subregion R_i is a polygon that has its geometric centre – the centroid c_i. The centroid is the "centre of mass" of the shape, and it is calculated based on the geometry of the polygon. Together with the initial entry point of the swarm p_0, which is given as coordinates (x_0, y_0), a set of points is formed that must be visited sequentially.

To determine which point to visit next, we need to know the distances between all pairs of points. In this case, the Euclidean distance *d* is used, which for any pair of points $p_i = (x_i, y_i)$ and $p_j = (x_j, y_j)$ is calculated by the formula:

$$d\left(p_i, p_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}\ .$$

These distances are written in the form of a square distance matrix, where each element indicates the distance between specific points.

The greedy algorithm starts from the initial point p_0, which is the entry point for the swarm into the scanning area, and then successively selects the nearest centroid that has not yet been visited. At each step, the algorithm compares the distances to all available options and makes a locally optimal choice, *i.e.*, moves to the point that is at the minimum distance from the current location.

For example, if the last visited point was the centroid c_k, the next one will be the one c_j with the smallest value (c_k, c_j) among the points not yet visited.

This approach does not guarantee an absolute global minimum of the entire route length, as an exact algorithm would, but it greatly simplifies the calculations and gives a result sufficient for practical applications. When all centroids have been visited, the swarm returns to the starting point p_0, forming a complete cycle.

OPTIMISE SCANNING OF DEFINED SUBREGIONS

It is not enough to simply divide the area into subregions and optimise the order of their scanning. It is also important to efficiently distribute the area to be scanned among the swarm members, which may have different scanning widths and speeds depending on the recipe that was set in the previous stages of work.

Let n be the number of drones, and their respective coverage widths are denoted as $s_1, s_2, \ldots, s_i$ where i is the number of recipes with different scan widths. The idea is to divide the region *P* into bands, the width of each of which is proportional to the scan width and speed of the drones and form a zigzag route in each of these bands.

In addition, it is worth noting that during the task, even along pre-calculated routes, the actual movement of the drones is not completely deterministic. Due to the peculiarities of swarm coordination using swarm chemistry, the trajectories of individual agents may deviate from the planned ones – the so-called 'drift'. This is a natural part of the model that avoids excessive rigidity in movement and considers the dynamics of the environment.

However, to maintain swarm integrity and avoid group disintegration, swarm chemistry contains stabilisation mechanisms that constantly monitor the distances between drones. When one or more drones start moving too far away from the main swarm (*i.e.*, beyond the communication or listening range), the system activates corrective forces of mutual attraction, forcing such agents back within the acceptable limits.

Thus, at the first stage of building a path for swarm agents, the boundaries of the subregion covered by R are determined:

$$\left[x_{min}, y_{min}, x_{max}, y_{max}\right] = \text{bounds}\left(R\right).$$

 ISSN 2617-4316 (Print)
ISSN 2663-7723 (Online)

The full width W of the area is determined based on these boundaries:

$$W = x_{\max} - x_{\min} .$$

Based on these bounds, the full height H of the subregion is computed as:

$$H = y_{\max} - y_{\min} ,$$

Next, the total sum of the coverage widths of all swarm agents is calculated:

$$T = \sum_{i=1}^{n} w_i ,$$

where T is total effective scanning width of all agents in the swarm; n is number of drones in the swarm; w_i is individual scanning width of the i-th drone.

This value is used as the basis for proportional space allocation. For each drone i, its share in the total width of the area is determined:

$$p_i = \frac{w_i}{T} .$$

Accordingly, the width of the band to be covered by drone i is calculated as:

$$s_i = W \cdot p_i = W \cdot \frac{w_i}{T} .$$

Each such strip is created as a rectangle with sides from x_{current} to $x_{\text{current}} + s_i$, where x_{current} is the coordinate of the start of the next strip (starting from $x_{\min}$). Since the polygon R can be of a complex shape, not just a rectangle, the actual working area of each drone is defined as the intersection of its rectangular strip with the polygon:

$$R_i = R \cap \left[x_{\text{current}}, y_{min}, x_{\text{current}} + s_i, y_{max} \right]$$

This ensures that none of the drones operate outside the main polygon. Once each subregion is defined, a zigzag scan route is generated. The entire workflow is shown in Fig. 3.

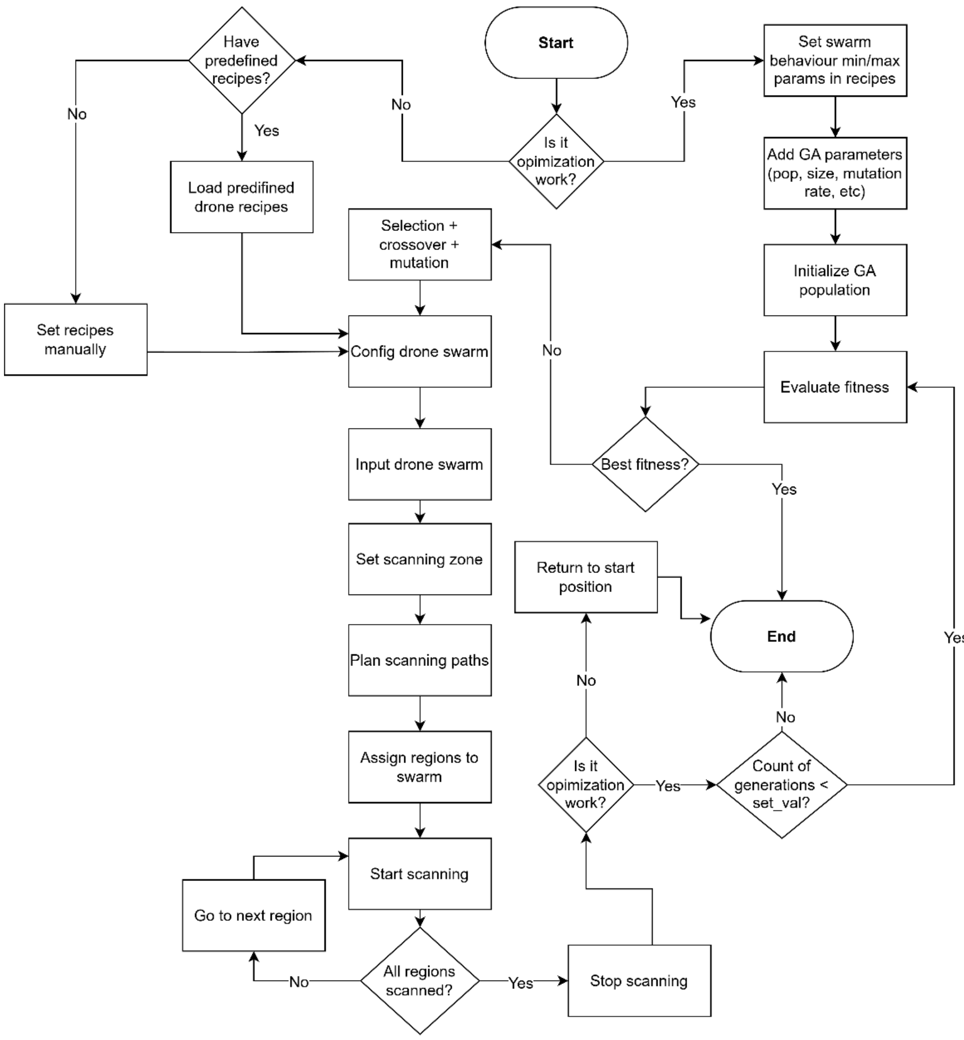

Fig. 3. **Flowchart of the hybrid algorithm for reconfiguring the swarm complex**
***Source*: compiled by the authors**

EXPERIMENTS

The simulation was carried out on the territory defined as a pentagon. All computations and modelling are implemented in Python. Before the main scan was launched, a preliminary training stage was conducted using a genetic algorithm. The purpose of this stage was to determine the optimal set of drone parameters, including the scanning width, as well as the minimum and maximum speed, selected within the given recipes. In this way, the algorithm selects parameter values that fully match the technical capabilities of the drones.

The training lasted for 48 generations (Fig. 4). At each step, the fitness of individuals was assessed based on a comprehensive criterion that included the efficiency of covering the area with minimal gaps and overlaps, and the total time of the mission. It is important to note that a successful scan was one with a coverage rate of at least 95 %, although this threshold is configurable.

The use of selection, crossover and mutation contributed to the gradual improvement of parameters with each generation which allowed us to achieve a balance between the quality of coverage and minimisation of operation time. After completing the training phase, the main simulations of territory scanning were conducted with the parameters of the recipes obtained after the genetic algorithm. An example of such a simulation is shown in Fig. 5.

In this figure, we can see the division of the initial pentagonal zone into smaller subregions, each of which has its own number, reflecting the order of scanning by the swarm of subregions determined by a greedy algorithm that takes into account minimising the total distance between zones. The simulation also includes a display of the scanned area.

Depending on the number of cells passes in the zone, the colour of the cell changes from cold (blue) to warm (red), which visually reflects the intensity of repeated scanning of this area of the territory (Fig. 5). This approach makes it easy to identify areas with more passes and optimise routes to reduce duplicate passes in the future, thereby increasing the overall efficiency of the scanning process. Moreover, the heatmap representation provides an intuitive tool for analysing the spatial distribution of drone activity, enabling rapid adjustments to scanning strategies during subsequent missions. The heatmap data can also support long-term analysis across multiple missions.

However, a slight duplication of scanning in real aerial reconnaissance conditions will allow for a more detailed coverage of the territory, which can potentially have a positive impact on the identification of objects of interest. In particular, the redundancy created by multiple passes reduces the risk of missing critical details due to temporary occlusions or environmental noise, ultimately enhancing the reliability of object detection in complex scenarios.

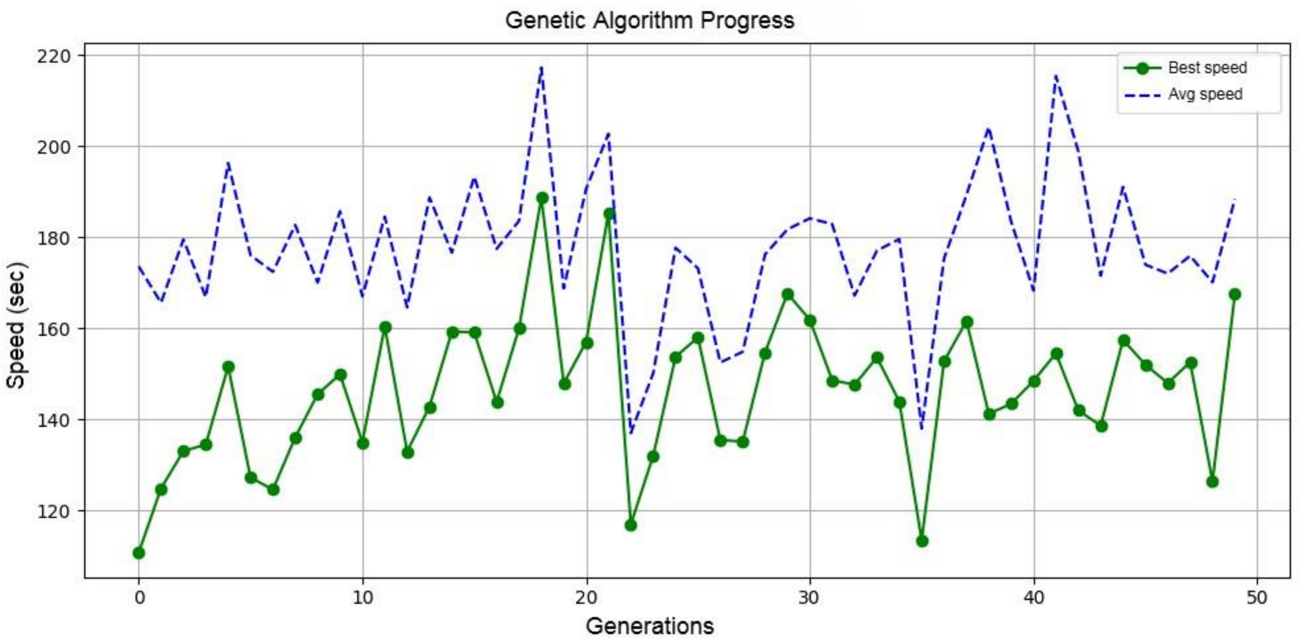

Fig. 4. **Diagram of scanning speed evolution with generations of the genetic algorithm**
***Source*: compiled by the authors**

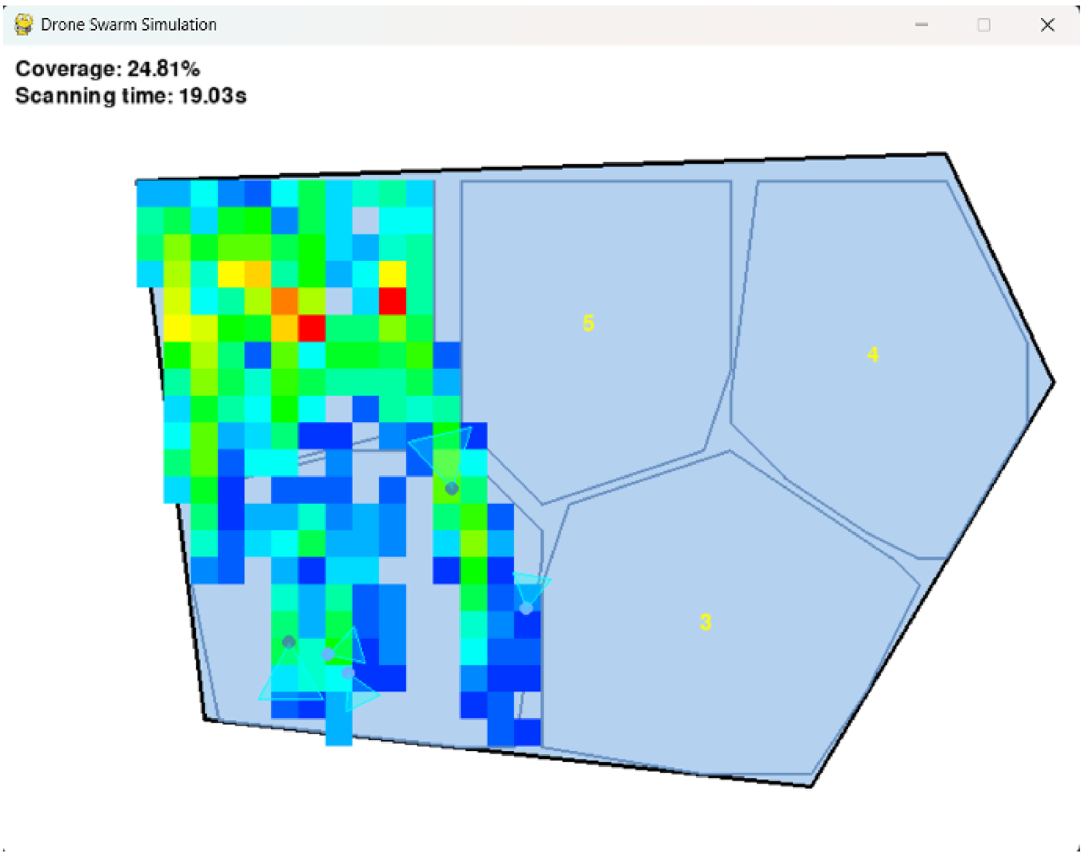

Fig. 5. **Simulation of swarm scanning of a specified area**
***Source*: compiled by the authors**

RESULTS

To be able to evaluate the effectiveness of the proposed approach, a series of preliminary simulations were conducted for a swarm consisting of five agents.

In the first experiment, the route optimisation technology was not used. Fig. 6 shows the results of 10 simulations of such flights.

The average scan time was 175 seconds, but there is considerable variability in the results: the fastest scan took 155 seconds and the slowest took 205 seconds, indicating that the results are unstable and that some "successful" trajectories are random rather than systematic.

In the second set of experiments, we used the above approach to optimise routes for a heterogeneous swarm of five agents.
The area to be scanned remained unchanged, and routes were formed based on a genetic algorithm considering the swarm heterogeneity and, accordingly, optimising the scanning area. The results of ten simulations are shown in Fig. 7. In this case, the average scanning time was significantly reduced to 82.77 seconds, which demonstrates a significant improvement in performance. In addition, the variability of the scanning time decreased: the values ranged from decreased: the values ranged from 75 to 90 seconds (Fig. 7), which indicates the stability and reliability of the optimised routes. This predictability allows for better mission planning and more efficient use of resources.

CONCLUSIONS

This paper discusses the technology of reconfiguring swarm complexes of unmanned aerial vehicles based on a combination of swarm chemistry and genetic algorithms. These two components – "chemistry" and "genetics" – have become the key drivers of the efficiency and relevance of the presented technology. Swarm chemistry provides dynamic interaction between agents, allowing them to form complex collective behavioural patterns based on local rules similar to chemical reactions between particles.

The genetic algorithm, in turn, acts as an evolutionary mechanism that makes it possible to find the best swarm configuration within the given recipes.

A special role was played by dividing the surveillance area into subregions and optimising their scanning order, which significantly increased the efficiency of territory coverage and reduced duplication of routes between agents.

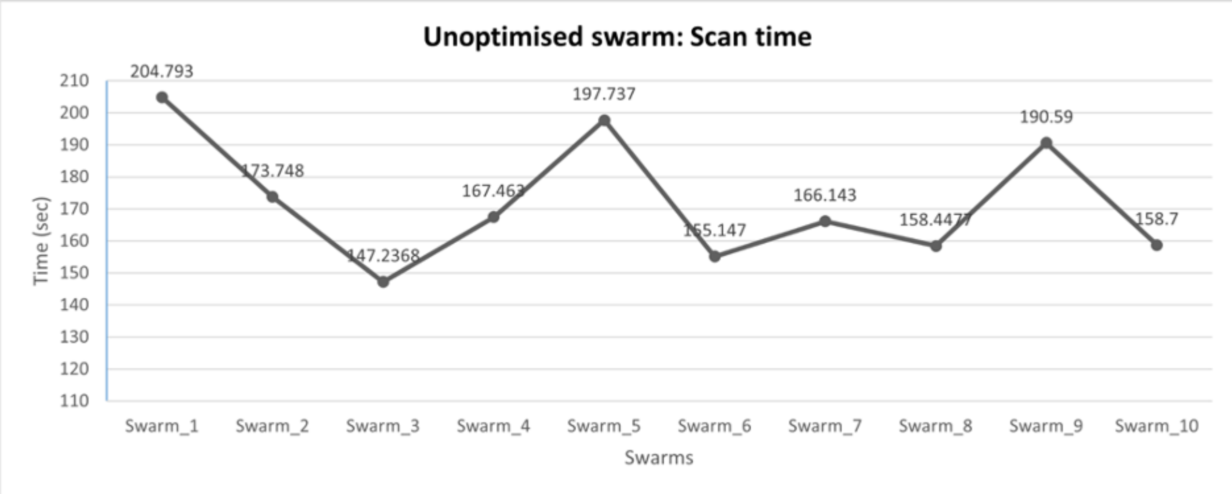

Fig. 6. **Diagram of swarm scanning time without optimisation**
***Source*: compiled by the authors**

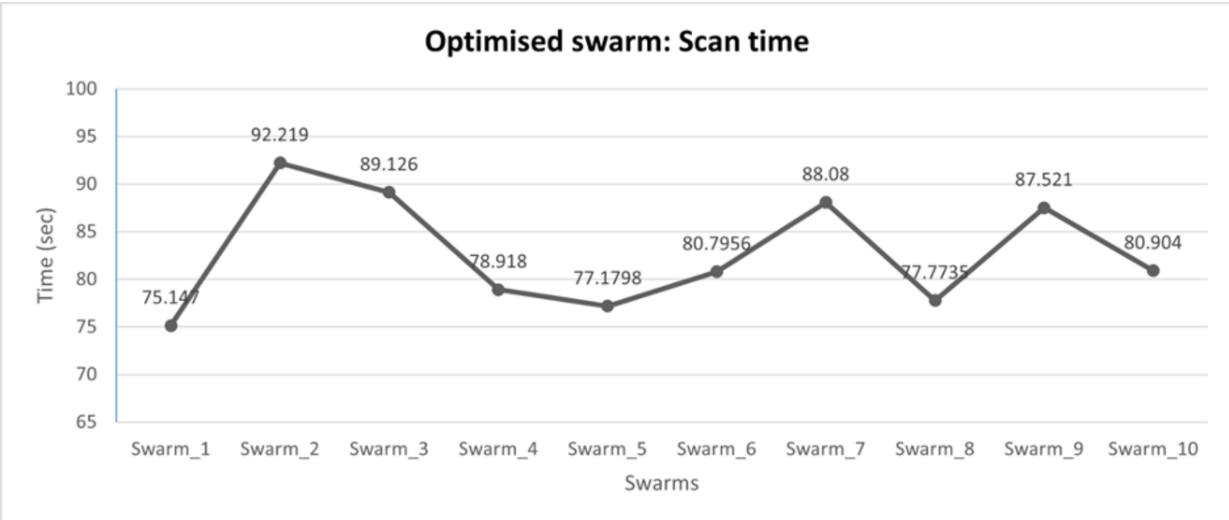

Fig. 7. **Diagram swarm scanning time after optimisation**
***Source*: compiled by the authors**

The simulation results confirmed the effectiveness of the proposed technology. Compared to the classical approach without optimisation, where the average scanning time was 175 seconds, the use of technology based on swarm chemistry and genetic algorithms reduced this figure to 82.77 seconds. Thus, the scanning time was reduced by 2.1 times, which is a significant achievement.

In addition to reducing the mission duration, a significant reduction in the variability of the results was also achieved – the time spread between the worst and best flight was reduced, indicating the stability and predictability of swarm behaviour. This is especially important in practical applications where the reliability of the task is a critical factor.In the future, the technology has even greater potential due to the possible introduction of reinforcement learning methods. This approach will allow agents to learn from previous experience, as proposed in [23], [24], [25], [26] to adapt their behaviour to new conditions and automatically improve strategies based on feedback. To enhance the autonomy of swarm agents in dynamic scenarios where the speed of decision-making and learning is critical, the NF-RBFNN approach should be analysed [27]. The combination of swarm chemistry, genetic algorithms, and reinforcement learning is expected to provide the foundation for a fully adaptive, self-learning, and efficient swarm capable of operating in a complex and changing environment. Another important step will be the transition to 3D simulation for greater realism and the possibility of more diverse testing of the technology.

REFERENCES

1. Banga, A., Ahuja, R. & Sharma S. C. "Human activity recognition: Approaches, datasets, applications, and challenges". *Modern Approaches in IoT and Machine Learning for Cyber Security*. 2023; 17 (2): 103–112, https://www.scopus.com/sourceid/21100826426. DOI: https://doi.org/10.1007/978-3-031-09955-7_7.

2. Tsariuk, Andrey O. & Malakhov, E. V. "The multilayer distributed intelligence system model for emergency area scanning". *Information Technology in Socio-Economic, Organizational and Technical Systems*; 202; 14 (3): 268–277. DOI: https://doi.org/10.15276/hait.03.2021.6.

3. Yucesoy, E., Balcik, B. & Coban, E. "The role of drones in disaster response: A literature review of operations research applications". *International Transactions in Operational Research*. 2024; 32:545–589, https://www.scopus.com/sourceid/9700153238. DOI: https://doi.org/10.1111/itor.13484.

4. Samburskyi, V. O., Malakhov, E. V. "Improving the survivability of a reconnaissance UAV swarm based on cellular automata". *Association for Computing Machinery*. 2024; 13 (11): 310–315, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85204938525&doi=10.1145%2f3674558.3674603&partnerID=40&md5=e00af8b6f5f4107cee22548dd3c735f1. DOI: https://doi.org/10.3390/electronics13112005.

5. Sieber, C., Vieira da Silva, L. M., Grünhagen, K. & Fay, A. "Rule-Based Verification of Autonomous Unmanned Aerial Vehicles". *Drones*. 2024; 8 (1): 26, https://www.scopus.com/sourceid/21101017244. DOI: https://doi.org/10.3390/drones8010026.

6. Yin, S., Jin, M., Lu, H. et al. "Reinforcement-learning-based parameter adaptation method for particle swarm optimization". *Complex Intelligent Systems*. 2023: 9: 5585–5609, https://www.scopus.com/sourceid/21101095931. DOI: https://doi.org/10.1007/s40747-023-01012-8.

7. Sengupta, S., Basak, S. & Peters, R. A., II. "Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives". *Machine Learning and Knowledge Extraction*. 2019; 1 (1): 157–191, https://www.scopus.com/sourceid/21101109601.
DOI: https://doi.org/10.3390/make1010010.

8. Tian, Y., Zhang, J., Wang, Q. et al. "Application of Hybrid Algorithm Based on Ant Colony Optimization and Sparrow Search in UAV Path Planning". *International Journal of Computational Intelligence Systems*. 2024; 17 (286), https://www.scopus.com/sourceid/17800156712. DOI: https://doi.org/10.1007/s44196-024-00652-z.

9. Gao, C., Ma, J. & Li, T. "Hybrid swarm intelligent algorithm for multi-UAV formation reconfiguration". *Complex Intelligent Systems*. 2022; 9 (9): 1929–1962, https://www.scopus.com/sourceid/21101095931. DOI: https://doi.org/10.1007/s40747-022-00891-7.

10. Kozlov, M. S. & Malakhov, E. V. "A Complex Model of Human Recognition in Emergencies by a Multi-level Drone System". *2024 International Symposium ELMAR*. 2024; p. 317–320, https://www.scopus.com/sourceid/145063. DOI: https://doi.org/10.1109/ELMAR62909.2024.10694255.

11. Westheider, J., Rückin, J. & Popovic, M. "Multi-UAV Adaptive Path Planning Using Deep Reinforcement Learning". *IEEE/RSJ International Conference on Robotics and Intelligent Systems*; 2023. DOI: https://doi.org/10.48550/arXiv.2303.01150.

12. Penko, O., Pienko, V. & Malakhov, E. "Multiagent Path Finding using Dijkstra Algorithm". *IEEE 19th International Conference on Computer Science and Information Technologies (CSIT)*; 2024. p. 1–4, https://www.scopus.com/sourceid/21101058059. DOI: https://doi.org/10.1109/CSIT65290.2024.10982587.

13. Wang, P., Su, G., Yang, W. & Jing, P. "Research on Safety Interlock System Design and Control Experiment of Combined Support and Anchor Equipment". *Sensors*. 2022; 22 (16), https://www.scopus.com/sourceid/130124. DOI: https://doi.org/10.3390/s22166058.

14. Wang, Y., Xing, L., Wang, J., Xie, T. & Chen, L. "Multi-Objective Rule System Based Control Model with Tunable Parameters for Swarm Robotic Control in Confined Environment". *Complex System Modeling and Simulation*, 2024; 4 (1): 33–49, https://www.scopus.com/sourceid/21101112543. DOI: https://doi.org/10.23919/CSMS.2023.0022.

15. Lin, S., Wang, J., Huang, B. et al. "Bio particle swarm optimization and reinforcement learning algorithm for path planning of automated guided vehicles in dynamic industrial environments". *Scientific Reports*. 2025; 15 (463), https://www.scopus.com/sourceid/21100200805.
DOI: https://doi.org/10.1038/s41598-024-84821-2.

16. Shiyuan, Y. “Reinforcement-learning-based parameter adaptation method for particle swarm optimization”. *Complex & Intelligent Systems*. 2023; 9 (5), https://www.scopus.com/sourceid/4900152708. DOI: https://arxiv.org/abs/2206.00835.

17. Zhang, Y., Ding, M., Zhang, J. et al. “Multi-UAV pursuit-evasion gaming based on PSO-M3DDPG schemes”. *Complex Intelligent Systems*. 2024; 10: 6867–6883, https://www.scopus.com/sourceid/21101095931. DOI: https://doi.org/10.1007/s40747-024-01504-1.

18. Phan, A. “Particle swarm based reinforcement learning for path planning and traffic congestion”. *University of Alabama Libraries*. 2022. – Available from: https://ir.ua.edu/items/a791c75d-fb7b-49b7-a654-04d8dafaa008.

19. Sayama, H. “Swarm Chemistry”. *Artificial Life*. 2009; 15 (1): 105–114, https://www.scopus.com/sourceid/24142. DOI: https://doi.org/10.1162/artl.2009.15.1.15107.

20. Sayama, H. “Seeking Open-Ended Evolution in Swarm Chemistry II: Analyzing Long-Term Dynamics via Automated Object Harvesting”. *ALIFE*. 2020. p. 59–66, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85084727929&partnerID=40&md5=50ac582648acbfdb060286b0ebe7925d. DOI: https://doi.org/10.48550/arXiv.1804.03304.

21. Sayama, H. & Wong, C. “Quantifying evolutionary dynamics of swarm chemistry”. *ECAL 2011*. 2011. p. 729–730, https://www.scopus.com/inward/record.uri?eid=2-s2.0-85153093581&doi=10.7551%2f978-0-262-29714-1 ch110&partnerID=40&md5=25d43b5270e3eb491f601760e9565f39. DOI: https://doi.org/10.7551/978-0-262-29714-1-ch110.

22. MacQueen, J. B. “Some methods for classification and analysis of multivariate observations”. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*. 1967; 1: 281–297. – Available from: http://projecteuclid.org/euclid.bsmsp/1200512992.

23. Kalinina, I., Bidyuk, P., Gozhyj, A., Malchenko, P. “Combining forecasts based on time series models in machine learning tasks”. *CEUR Workshop Proceedings*. 2023; 3426: 25–35, https://www.scopus.com/sourceid/21100218356. – Available from: https://ceur-ws.org/Vol-3426/paper3.pdf.

24. Alharbi, A .M., Alshehri, G. & Elhag, S. “Reinforcement learning of emerging swarm technologies: A literature review”. *Proceedings of the Future Technologies Conference (FTC)*. 2024; 1156: 478–494, https://www.scopus.com/sourceid/21100901469. DOI: https://doi.org/10.1007/978-3-031-73125-9_31.

25. Huttenrauch, M., Socic, A. & Neumann G. “Deep reinforcement learning for swarm systems”. *Journal of Machine Learning Research*. 2019; 20 (54): 1–31. DOI: https://doi.org/10.48550/arXiv.1807.06613.

26. Wu, Q., Liu, K., Chen, L. & Lu, J. “Hierarchical reinforcement learning for swarm confrontation with high uncertainty”. *IEEE Transactions on Automation Science and Engineering*. 2025; 22: 8630–8644, https://www.scopus.com/sourceid/17340. DOI: https://doi.org/10.48550/arXiv.2406.07877.

27. Bodyanskiy, Y., Chala, O., Filatov, V. & Pliss I. “Neo-Fuzzy Radial-Basis function neural network and its combined learning”. *System Analysis and Artificial Intelligence*. 2023; 1107:3 23–340, https://www.scopus.com/sourceid/4900152708. DOI: https://doi.org/10.1007/978-3-031-37450-0_19.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 10.04.2025
Received after revision 12.06.2025
Accepted 18.06.2025

DOI: https://doi.org/10.15276/aait.08.2025.10
УДК: 004.89

Багаторівнева реконфігурація гетерогенного ройового комплексу на основі методу Swarm Chemistry

Козлов Микита Сергійович[1)]
ORCID: https://orcid.org/0009-0007-0071-7748; nikita.kozlov@stud.onu.edu.ua
Швець Юлія Олександрівна[1)]
ORCID: https://orcid.org/0009-0008-1714-3127, yulia.shvets@stud.onu.edu.ua
[1)] Одеський національний університет імені І. І. Мечникова, Дворянська, 2, Одеса, 65082, Україна

АНОТАЦІЯ

Необхідність оперативного та безпечного аналізу зон, що постраждали від надзвичайних ситуацій, спонукає до пошуку інноваційних підходів у сфері автономних систем. Одним із таких рішень є застосування ройових систем безпілотних літальних апаратів для сканування територій. У цьому дослідженні представлено гібридний підхід до динамічного управління роєм дронів, який поєднує механізми самоорганізації на основі методу «ройової хімії» з глобальною оптимізацією параметрів за допомогою еволюційного алгоритму. Запропоновано модуль багаторівневої реконфігурації рою, що дозволяє адаптувати поведінку як окремих агентів, так і всієї формації у відповідь на зміни навколишнього середовища. Реалізовано модуль кластеризації для поділу області сканування на підобласті, а також систему маршрутного планування з урахуванням індивідуальної ширини сканування кожного дрона. Для ефективного розподілу зон між агентами використано пропорційний алгоритм, що враховує функціональні можливості кожного апарата. Результати моделювання засвідчили понад дворазове скорочення часу виконання місії та підвищення стабільності поведінки рою, навіть за умов неоднорідності його складу та часткової втрати агентів. Представлений підхід має значний прикладний потенціал у сфері моніторингу надзвичайних ситуацій, пошуку постраждалих та оцінки масштабів руйнувань. У дослідженні сформульовано архітектуру динамічної реконфігурації рою, здатну адаптуватися до змін середовища в режимі реального часу. Такий підхід забезпечує гнучкість і стійкість системи під час виконання завдань у складних умовах. Подальший розвиток моделі передбачає інтеграцію методів машинного навчання для підвищення рівня адаптивності, а також розширення до тривимірного простору задля досягнення більшої точності й реалізму симуляції.

Ключові слова: інформаційні технології; ройовий інтелект; генетичний алгоритм; мультиагентні системи; оптимізація сканування; Swarm Chemistry

ABOUT THE AUTHORS

Mykyta S. Kozlov – PhD student of Computer Science. Odesa I. I. Mechnikov National University, 2, Dvoryanskaya Str. Odesa, 65082, Ukraine. Senior Software Development Engineer at SPD-Tech, Cherkasy, Ukraine.
ORCID: https://orcid.org/0000-0001-8130-9613; nikita.kozlov@stud.onu.edu.ua
Research field: Computer vision, swarm technology, artificial intelligence

Козлов Микита Сергійович – аспірант Комп'ютерних наук. Одеський національний університет імені І. І. Мечникова, Дворянська, 2, Одеса, 65082, Україна. Senior Software Development Engineer (Back-End), SPD-Tech, Черкаси, Україна

Yuliia O. Shvets – Master's student of Information Systems and Technologies. Odesa I. I. Mechnikov National University, 2, Dvoryanskaya Str. Odesa, 65082.
ORCID: https://orcid.org/0009-0008-1714-3127; yulia.shvets@stud.onu.edu.ua
Research field: Computer vision, swarm technology, artificial intelligence

Швець Юлія Олександрівна – магістр з Інформаційних систем та технологій. Одеський національний університет імені І. І. Мечникова, Дворянська, 2, Одеса, 65082, Україна