A hybrid approach based on swarm intelligence and behavior trees for coordinating autonomous agents

Michael Zgurovsky[1,†], Yuriy Zaychenko[2,*,†], Helen Zaichenko[2,†] and Oleksii Kuzmenko[2,†]

[1] *Educational and Scientific Complex "Institute for Applied System Analysis" of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Prospect Beresteiskyi, 37, 03056, Kyiv, Ukraine*

[2] *Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Prospect Beresteiskyi, 37, 03056, Kyiv, Ukraine*

Abstract

In this paper, a hybrid approach for coordinating autonomous agents is proposed, combining swarm intelligence based on the GBestPSO (Global Best Particle Swarm Optimization) algorithm and behavior trees (BTs). This approach aims to solve the problem of balancing global coordination of actions and local autonomy of each agent in dynamic and uncertain environments. The proposed two-level control system architecture consists of a planning level (GBestPSO-Level) for global optimization and a behavior level (BT-Level) for tactical behavior. The results were partially supported by the National Research Foundation of Ukraine, grant No. 2025.06/0022 "AI platform with cognitive services for coordinated autonomous navigation of distributed systems consisting of a large number of objects".

Keywords

multi-agent systems, drone swarm, autonomous systems, hybrid algorithms, self-organization

1. Introduction

Modern systems with collective interaction of autonomous agents are attracting increasing attention from researchers due to their ability to solve complex tasks in environments with a high level of uncertainty [1]. In such systems, agents function in a decentralized manner, exchange information, and jointly achieve goals, which makes them suitable for use in reconnaissance, monitoring, search and rescue, logistics, facility security, and other fields [2]. The key challenge in this context is to develop coordination methods that can ensure a balance between global coordination of actions and local autonomy of each agent.

Traditional centralized approaches, which assume the existence of a single controller, often prove ineffective in dynamic environments where there is a possibility of communication loss, system failure, or the emergence of new obstacles. In contrast, decentralized methods inspired by natural models of collective behavior, such as swarms of insects or flocks of birds, demonstrate significantly greater resilience and flexibility. One of the most common tools in this class is the Particle Swarm Optimization (PSO) algorithm, which allows agents to collectively find effective solutions by exchanging local and global information [3].

However, PSO alone does not provide a high level of cognition for an individual agent. It allows determining the direction and trajectory of movement, but does not provide a flexible mechanism for decision-making in situations where environmental conditions, unforeseen events, or complex

Information Technology and Implementation (IT&I-2025), November 20-21, 2025, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

zgurovsm@hotmail.com (M. Zgurovsky); zaychenkoyuri@ukr.net (Yu. Zaychenko); zaichenko.helen@lll.kpi.ua (He. Zaichenko); oleksii.kuzmenko@ukr.net (O. Kuzmenko)

0000-0001-5896-7466 (M. Zgurovsky); 0000-0001-9662-3269 (Yu. Zaychenko); 0000-0002-4630-5155 (He. Zaichenko); 0000-0003-1581-6224 (O. Kuzmenko)

interaction scenarios must be considered. In this context, the use of behavior trees (BT) [4] is of considerable interest. This approach, widely used in robotics and the gaming industry, provides modularity, hierarchy, and simplicity in describing agent behavior [5]. Using the BT structure, it is easy to integrate conditions, sequences of actions, and parallel processes, which allows for the creation of flexible and adaptive control algorithms.

The combination of PSO and BT forms the basis of a hybrid approach in which swarm optimization is responsible for global coordination and trajectory optimization, while behavior trees are responsible for local decision-making and tactical responses of agents [6]. This approach provides a double level of stability: on the one hand, the system's ability to achieve common goals is preserved even with the loss of individual agents or communication channels; on the other hand, each agent has a sufficient level of autonomy to act in complex and dynamic conditions.

This work pays particular attention to the application of a modified GBestPSO algorithm, which is supplemented by mechanisms of self-organization and adaptation to the influence of external factors that disrupt coordination [7]. In combination with behavior trees, this approach allows the creation of an architecture capable of solving both strategic tasks at the group level and tactical tasks at the individual agent level.

To verify the effectiveness of the proposed approach, two application scenarios are considered. The first is a swarm attack using the “death ring” pattern, which demonstrates the hybrid system's ability to coordinate a collective attack, ensuring synchronization of actions and achievement of the goal. The second is reconnaissance of enemy territory, illustrating the effectiveness of the method in space allocation tasks and avoiding duplication of actions while maintaining global coordination. Both scenarios confirm that the integration of PSO with behavior trees creates the basis for a more robust and flexible control architecture in multi-agent systems.

Thus, the research aims to substantiate and demonstrate the capabilities of a hybrid approach combining swarm intelligence and behavior trees as a promising direction for the development of technologies for coordinating autonomous agents in complex and dynamic environments.

2. Related works

Research into the collective behavior of autonomous agents combines biological inspiration, optimization methods, and engineering approaches. Many models are based on the idea of self-organization, borrowed from observations of animal-flocks of birds, colonies of ants, or swarms of insects [8]. These principles were subsequently formalized in the form of optimization algorithms and coordination models, which are now actively used in robotics, multi-agent systems, and distributed control systems.

One of the most popular paradigms is Particle Swarm Optimization (PSO), proposed by Kennedy and Eberhart in 1995 [3]. It models the process of finding the optimal solution through the dynamics of a swarm of particles moving in the search space, guided by their own experience and the successes of their neighbors. The GBestPSO modification focuses on the global leader – the most successful particle that determines the direction of development of the entire group [7]. This increases the convergence speed but makes the system more vulnerable to local minimum and dependent on a single center of influence.

Other swarm algorithms, such as Ant Colony Optimization (ACO) [9] and Bee Colony Optimization (BCO) [10], offer different mechanisms for collective decision-making. ACO is widely used in routing and logistics problems due to its efficiency in discrete spaces. Its key advantage is the use of the phenomenon of stigmergy – indirect interaction through changes in the environment (e.g., pheromone trails) [11]. However, the algorithm requires numerous iterations and significant resources to maintain global consistency, which reduces its suitability in real-time scenarios.

Further modifications of swarm algorithms, such as Firefly Algorithm [12], Cuckoo Search [13], or Glowworm Swarm Optimization [14], have been proposed to better balance global and local search capabilities. They demonstrate high efficiency in theoretical tests but are rarely used in practical robotic systems due to the complexity of parameterization and the lack of proven on-board implementations.

In the early stages of multi-agent system development, simpler approaches were actively used. The Boids model, proposed by Reynolds in 1987, was the first simulation of bird flock behavior based on three simple rules: alignment, collision avoidance, and attraction to the center of the group [15]. Despite its simplicity, this approach is still used in simulations and computer graphics, but its limitation lies in the absence of a cognitive level – agents cannot make complex decisions or adapt to mission changes.

Another direction of development is consensus algorithms, which focus on achieving a coordinated state among all agents through iterative information exchange. They scale well and have proven effective in synchronization tasks but have limited functionality in dynamic environments where not only coordination, but also adaptive strategy change is required.

The leader-follower model, which places responsibility on a key agent, has similar problems [16, 17]. The loss of a leader or their temporary unavailability leads to a breakdown in coordination, making the architecture vulnerable.

Recent years have been marked by the active introduction of deep reinforcement learning (DRL) methods into multi-agent systems [18]. DRL allows agents to learn optimal strategies through trial and error, forming policies capable of generalizing new scenarios. Examples include Deep Q-Network (DQN) [19], Proximal Policy Optimization (PPO) [20], and Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [21] algorithms.

Despite its high potential effectiveness, DRL faces a few limitations. First, training requires millions of episodes, making it unsuitable for rapid deployment in the field. Second, the models are "black boxes," which complicates the verification and certification of autonomous systems. Third, high computational costs make them unsuitable for implementation on resource-constrained robots without powerful GPUs or TPUs [22].

Behavior trees (BTs) have become one of the most promising methods for building control architectures in robotics [4]. They successfully combine modularity, hierarchy, and flexibility, making them a natural development of finite state machines [23] and HTN (Hierarchical Task Networks) planners [24].

BTs allow you to describe behavior through a combination of control nodes and actions that form a tree with a clearly defined execution logic. Their key advantages are:

- ease of reusing individual subtrees in different tasks
- ability to easily extend the structure with new nodes without complete redesign
- absence of a single critical element, even with the loss of individual agents
- ability to rebuild behavior logic during mission execution.

In decentralized architectures, BTs allow each agent to make local decisions, coordinating with neighbors only at the level of exchanging minimal information. This makes them suitable for large-scale systems where centralized control is impossible.

A review of the literature shows that none of the approaches is universal. Swarm optimization algorithms are good at global planning and search, but do not provide flexible reactive behavior. Conversely, BTs make it easy to model local adaptability and cognitive autonomy but lack mechanisms for global optimization.

That is why modern research shows a tendency toward integrating different methods. For example, PSO or GBestPSO can determine the optimal location or distribution of tasks among agents, after which local execution is coordinated through BTs [6, 25]. This combination provides both global coordination and local autonomy, which is especially important in scenarios with a high level of uncertainty or when working in dynamic environments.

Thus, the current state of research in the field of collective behavior of autonomous agents can be characterized as a gradual transition from monolithic approaches to composite architectures that combine the strengths of different methods. Among them, the integration of GBestPSO for global search and BTs for local autonomy attracts particular attention. This approach not only solves the problems inherent in each of the methods separately but also forms a new level of stability and adaptability necessary for building multi-agent systems capable of functioning in real, unpredictable conditions.

3. Hybrid system architecture

Effective coordination of autonomous agents in a dynamic and resource-constrained environment requires a multi-level control system capable of combining global planning with local reactive behavior. The proposed architecture is based on the integration of two approaches: Behavior Trees (BTs), which provide modularity and flexibility at the tactical level, and the GBest Particle Swarm Optimization (GBestPSO) algorithm, which is responsible for the strategic coordination of the entire system.

The architecture has a two-level structure (Fig. 1): the planning level (GBestPSO-Level) and the behavior level (BT-Level).

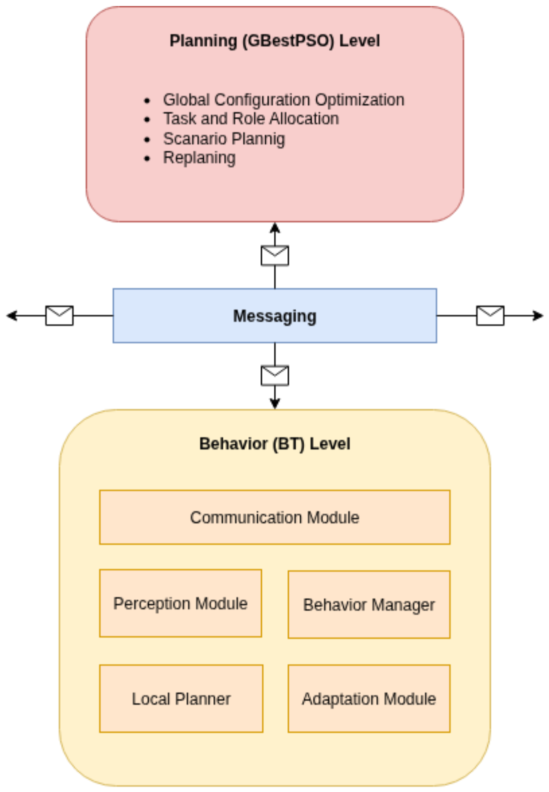

Figure 1: Two-Level Architecture of the Hybrid System

The planning level (GBestPSO-Level) acts as a global optimizer that forms the strategic basis of the mission. Its main functions are:

- Optimization of the global configuration of the swarm. The GBestPSO algorithm periodically calculates the best location of agents in space, considering goals and constraints.
- Distribution of tasks and roles. Each agent is assigned a subtask that corresponds to its capabilities and the strategic goals of the system.
- Scenario planning. In complex missions, it is possible to form several scenarios of events with the subsequent selection of the optimal one.

- Replanning. In the event of significant changes in the environment (for example, the emergence of new threats or obstacles), GBestPSO is restarted to correct the strategic plan.

This level can be viewed as the strategic “gravity center” of the system, which defines the goal of the group of agents and coordinates their actions. Unlike centralized approaches, the architecture allows GBestPSO to operate as “advice” that is periodically sent to agents, rather than as a real-time directive. This allows the system to remain stable even during temporary communication failures.

The behavior level (BT-Level) determines the individual tactical behavior of each agent, which is equipped with a behavior tree. This level includes the following modules:

- Behavior Manager. Performs real-time interpretation of the behavior tree, activates local actions, and rebuilds logic when conditions change.
- Perception Module. Provides sensor data processing, object recognition, and local environment mapping.
- Local Planner. Responsible for low-level navigation – obstacle avoidance, local SLAM, trajectory correction.
- Adaptation Module. Allows dynamic restructuring of the behavior tree in case of agent loss, role change, or communication interruption.
- Communication Module. Implements information exchange with other agents, including the transmission of states, positions, and parts of the behavior tree.

BT-level provides instant responsiveness to environmental events and allows agents to remain operational even when communication with the planning level is lost.

Thus, GBestPSO provides global coordination, while BTs ensure local autonomy. Their interaction is organized through standardized messaging protocols, allowing the system to remain functional even in the event of partial communication loss.

The connection between the planning and behavior levels is organized in such a way as to avoid agents' dependence on a single source of information. The algorithm can be described as follows: Swarm Optimizer (at the GBestPSO level) calculates the globally best strategy (gBest) and transmits it as a message, and BT-level perceives this information as external mission conditions, which are reflected in the tree branches.

In the event of a change in global strategy, the local behavior of agents is automatically adjusted. If communication with the planning level is lost, agents act based on the last received reference point, maintaining the ability to complete the mission. To distinguish between critical and auxiliary data, channels with different priority levels are used: messages about strategy changes have a higher weight than diagnostic telemetry. The interaction between levels is based on the principle of “weak coupling”: agents remain autonomous but can quickly integrate new information when it becomes available.

The proposed two-level architecture has some key advantages:

- Each agent can work without being constantly connected to the global planner.
- Behavior trees are easy to modify and scale for different missions.
- The system remains operational even if individual agents are lost or communications are temporarily disrupted.
- The architecture allows new algorithms to be integrated without the need for a complete system overhaul.
- It is suitable for a wide range of applications, from reconnaissance and escort to attack scenarios or search and rescue operations.

Thus, the architecture based on the combination of GBestPSO and BTs provides a balance between global optimization and local adaptability. It demonstrates the ability to work effectively in conditions of uncertainty, high environmental dynamics, and limited resources. The use of a hierarchical approach makes the system fault-tolerant, modular, and suitable for a variety of collective interaction scenarios.

4. Description of simulation scenarios and settings

Unmanned Aerial Vehicles (UAVs) were selected as autonomous agents for the study. The simulation model is based on a hybrid architecture that combines swarm optimization methods (GBestPSO) for dynamic modelling of drone trajectories and the Behavior Trees (BT) framework for implementing adaptive and fault-tolerant control logic. Each UAV functions as an autonomous agent controlled by its own behavior tree, which cyclically processes the current mission status and external influences.

4.1. Scenario 1. Swarm attack using the "death ring" pattern

This scenario simulates a coordinated attack by a swarm of drones on a stationary target, consisting of two consecutive phases: approach and formation of a ring of fire.

4.1.1. Phase 1: approach to the target

In the initial stage, the drones move toward the target using a modified swarm optimization algorithm (GBestPSO). The main difference between this algorithm and others is that the acceleration coefficients that regulate the influence of the leader (C_1) and the target (C_2) are not constants but are linearly dependent on the current distance of the drone from the corresponding object.

The equation for updating the speed of drone i in direction j is as follows:

$$\begin{gathered} v_{ij}(t+1) = v_{ij}(t) + C_1\left(d\big(Y_L(t), X_i(t)\big)\right) \cdot r_1(t) \cdot \left[Y_{L_j}(t) - X_{ij}(t)\right] + \\ + C_2\left(d\big(Y^*(t), X_i(t)\big)\right) \cdot r_2(t) \cdot \left[Y_j^*(t) - X_{ij}(t)\right], \end{gathered} \tag{1}$$

where coefficients C_1 and C_2 are calculated using the following formulas:

$$C_1\left(d\big(Y_L(t), X_i(t)\big)\right) = \frac{C_{1max} - C_{1min}}{\|Y_L(0) - X_i(0)\|} \cdot \left(Y_{Lj}(t) - X_{ij}(t)\right) + C_{1min} \tag{2}$$

$$C_2\left(d\big(Y^*(t), X_i(t)\big)\right) = \frac{C_{2max} - C_{2min}}{\|Y^*(0) - X_i(0)\|} \cdot \left(Y_j^*(t) - X_{ij}(t)\right) + C_{2min} \tag{3}$$

The UAVs follow these routes until the leader of the swarm (the UAV closest to the target) reaches the specified attack radius r_0.

4.1.2. Phase 2: firing by the ring

Once the leader reaches the target's vicinity with a radius of r_0 (100–200 m), the swarm enters the phase of rotation around it. Movement in this phase is described in a polar coordinate system relative to the target, which is considered the center.

The coordinates of the leader Y_L in the Cartesian system can be calculated based on its angular position $\varphi_L(t)$ and radius r_0:

$$\begin{cases} y_1^L(t) = r_0 \cdot cos\big(\varphi_L(t)\big) \\ y_2^L(t) = r_0 \cdot sin\big(\varphi_L(t)\big) \end{cases} \tag{4}$$

The angular position is updated at each step of the simulation using the following formula:

$$\varphi_L(t+1) = \varphi_L(t) + \omega_L(t)\Delta t \tag{5}$$

where $\omega_L(t)$ is the angular velocity of rotation.

The synchronization stage of the attack is critically important and is provided through the Decorator~ node in the Behavior Tree. This node allows the final attack to begin only after a certain minimum number of UAVs (M_{Y^*}) reach orbit, ensuring a simultaneous and effective strike.

The simulation settings for this scenario are shown in Table 1.

Table 1
Simulation Settings for Scenario 1

Parameter	Value	Description
Number of UAVs (N)	10-15	Total number of agents in the swarm.
Ring radius (r_0)	100-200 m	Radius of rotation of agents around the target.
Strike synchronisation	< 2 sec	Maximum permissible delay between attacks by individual UAVs.
Communication	ROS2 + DDS	Technologies that enable distributed and autonomous interaction.
Time step (Δt)	1-10 sec	Adjusted for simulation accuracy.
UAV velocity vector ($\vec{V_i}$)	10-50 m/s	Depends on the motion model and constraints.
Scale	1:100	One unit (tick) on the axis corresponds to 100 meters in the simulation.

4.2. Scenario 2. Reconnaissance of enemy territory

This scenario demonstrates the capabilities of a swarm for autonomous reconnaissance, where UAVs effectively divide the search area among themselves. The simulation scenario involves coordinating a swarm of autonomous agents to conduct reconnaissance of enemy territory. Each UAV is assigned a specific area for reconnaissance. The results from all areas are combined into a common map of the terrain, which significantly speeds up the process. The main goal is to ensure complete coverage of the designated area and localization of reconnaissance targets, considering possible UAV losses and route replanning.

4.2.1. Problem statement

Let the surveyed area be defined as a rectangular region with coordinates:

$$(x_{start}, y_{start}), (x_{end}, y_{end}) \tag{6}$$

The swarm consists of N agents, each of which receives an individual area for exploration. Each area is defined by a strip along the X-axis:

$$\Delta x = \frac{x_{end} - x_{start}}{N} \tag{7}$$

Coordinates of the i-th zone:

$$\begin{cases} x_{l,i} = x_{start} + (i-1)\Delta x \\ x_{r,i} = x_{start} + i\Delta x \\ y_{start,i} = y_{start}, y_{end,i} = y_{end}, \end{cases} \tag{8}$$

This data is transmitted to agents via a communication system.

4.2.2. Routes to reconnaissance areas

A hybrid GBestPSO+BTs algorithm is used to plan trajectories, where:

$$\begin{cases} X_i(k+1) = X_i(k) + v_{i1}(k)\Delta t_i \\ Y_i(k+1) = Y_i(k) + v_{i2}(k)\Delta t_i \end{cases} \tag{9}$$

where $\overrightarrow{V_i(k)} = (v_{i1}(k), v_{i2}(k))$ is the velocity vector of UAV i at iteration k, and Δt_i is the time step.

The swarm supervisor evaluates the coordinates of the target areas and optimizes the routes to avoid collisions and ensure minimum arrival time.

4.2.3. Movement in the reconnaissance zone

UAVs move along strips with a width of $d = 2r_0$, where r_0 is the observation radius of the UAV. Movement along the Y axis is defined by the equation:

$$y_i(k+1) = y_i(k) + v_i(k)\Delta t \tag{10}$$

After reaching the upper limit of the survey area, the UAV shifts along X by $\frac{d}{2}$ and returns in the opposite direction to survey the adjacent strip.

Conditions for completing the survey of the area:

$$x_i(k+1) + \frac{d}{2} \geq x_{r,i} \tag{11}$$

If some UAVs are lost or an area remains unexplored, the hybrid algorithm determines which UAVs should perform a re-survey. To do this, the position update rule is used:

$$v_i(k+1) = v_i(k) + C_1(k)r_1(x_{start,i} - x_i(k)) \tag{12}$$

where $C_1(k)$ is the cognitive influence coefficient, and $r_1 \sim U(0,1)$ is a random variable.

After completing the survey, the UAVs return to their initial coordinates (X_{i0}, Y_{i0}).

The simulation is configured using the parameters described in Table 2.

Table 2
Simulation Settings for Scenario 2

Parameter	Value	Description
Number of UAVs (N)	6-12	The selection depends on the scale of the area.
Reconnaissance strip width (d)	$2r_0$	Observation radius.
Time step (Δt)	1-10 sec	Adjusted for simulation accuracy.
Initial zone coordinates (x_{start}, y_{start})	-	Calculated using equation (7).
Final zone coordinates (x_{end}, y_{end})	-	Calculated using equation (7).
UAV velocity vector ($\vec{V_i}$)	10-30 m/s	Depends on the motion model and constraints.
Cognitive influence coefficient ($C_1(k)$)	0.5-2	Affects the PSO trajectory update.
Communication	ROS2 + DDS	Technologies that enable distributed and autonomous interaction.
Scale	1:100	One unit (tick) on the axis corresponds to 100 meters in the simulation.

5. Experimental investigations

The first series of experiments was conducted for the scenario “Swarm attack using the ‘death ring’ pattern” using only behavior trees (BT). The number of UAVs and movement parameter settings were varied. The main performance metrics were:

- time to attack initiation (number of iterations until the minimum number of UAVs entered orbit)
- total attack completion time (number of iterations until all UAVs targeted the object)
- UAV loss during approach and attack

- uniformity of UAV placement in orbit (average distance between UAVs in orbit).

During each experiment, the launch of UAV from a certain area of the arena was simulated (Fig. 2).

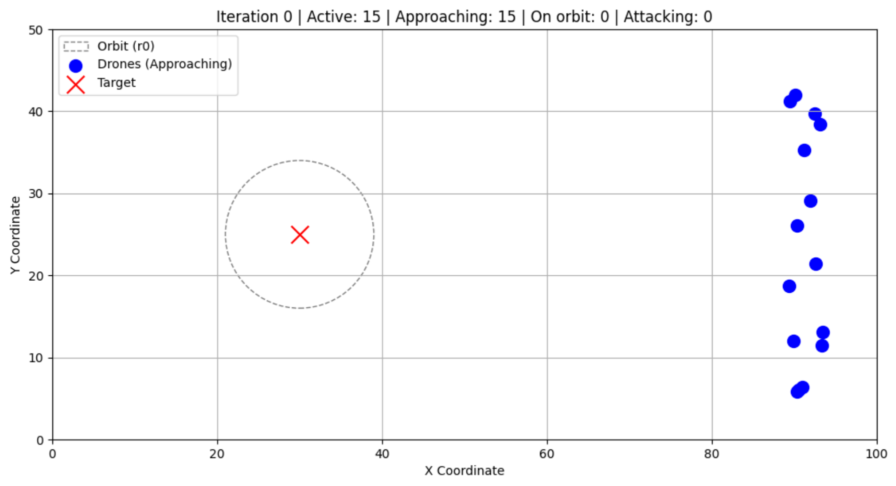

Figure 2: Launching UAVs from the Starting Position for Scenario 1 with only BT-based simulation

After launch, the UAVs were controlled by the internal BTs mechanism and moved to the target coordinates along the shortest trajectory (Fig. 3).

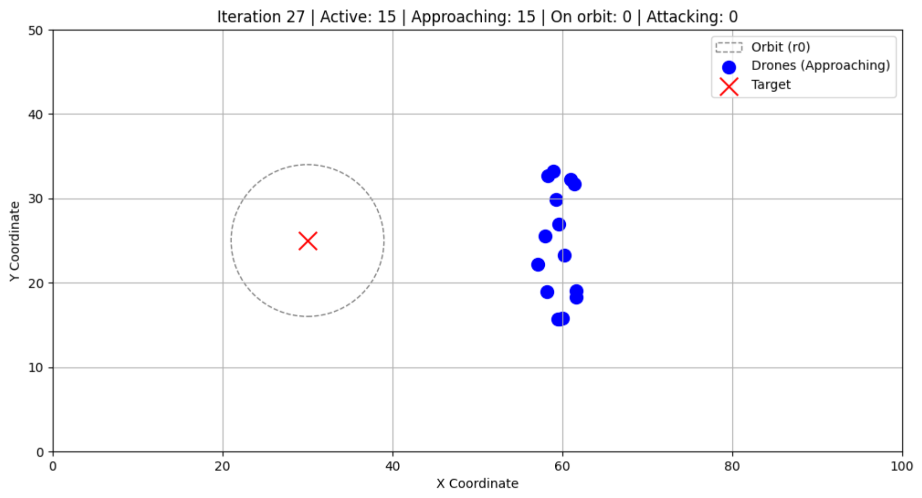

Figure 3: Movement of UAVs toward a target under the control of a BTs mechanism

Table 3
Main Simulation Results for the Scenario “Swarm Attack using the ‘Death Ring’ Pattern” using only Behavior Trees (BT)

Number of UAVs	Iterations before the start of the attack	Total attack iterations	UAV losses	Average distance between UAVs in orbit
10	45	112	1	9.2
15	32	97	2	8.7
20	28	85	3	8.5

The data shows that increasing the number of UAVs speeds up the start of the attack and reduces the time it takes to complete it due to faster formation of the orbit around the target. UAV losses remain low, indicating the stability of the behavior tree algorithm.

Table 4
UAV losses by iterations (example for 15 UAVs)

Iteration	Number of active UAVs	Number of UAVs in orbit	Number of attacking UAVs
0	15	0	0
20	15	4	0
32	15	10	5
50	14	7	7
97	13	0	13

The dynamics can be seen graphically in Figure 4, which shows the change in UAV states (approach, orbit, attack) over iterations.

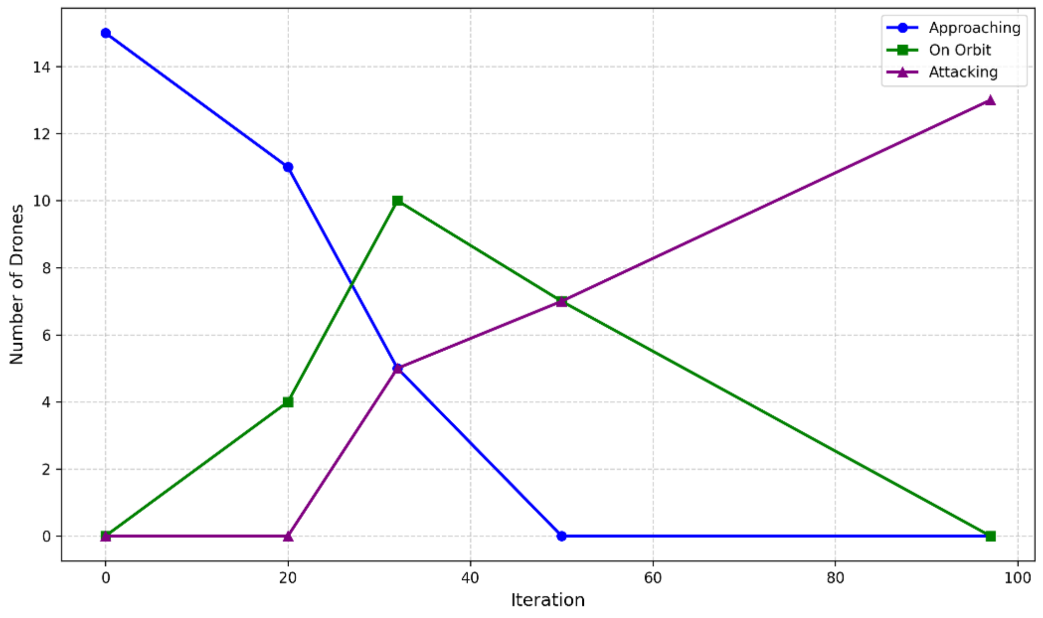

Figure 4: Dynamics of UAV State Changes (Approaching, On Orbit, Attacking) by iterations

The graph shows three stages of UAV behavior:

- Approaching – UAVs move toward orbit
- On Orbit – UAVs form a circle around the target
- Attacking – UAVs attack the target.

The next series of experiments was conducted to evaluate the effectiveness of the proposed hybrid method (GBestPSO+BTs) for a swarm attack scenario based on the “death ring” pattern. The swarm consisted of 15 UAVs that took off from an area located on the right side of the arena and had to coordinate and gather around the target at a specified point.

The purpose of the experiment was to measure the swarm's ability to:

- form a stable orbit around the target

- achieve a uniform angular distribution of UAVs in orbit
- initiate a coordinated attack when the appropriate conditions are met
- minimize UAV losses.

After the launch of the UAVs, the swarm leader was immediately determined (Fig. 5), which was at the shortest distance to the target's orbit.

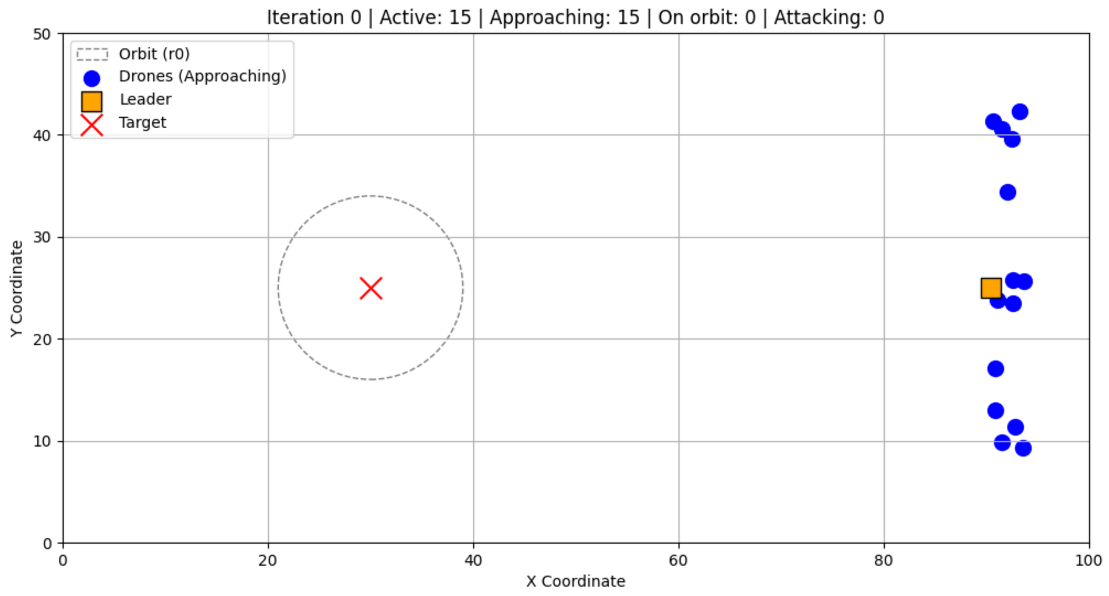

Figure 5: Selecting a Leader among UAVs at the Start of a Mission in a Simulation using the GBestPSO+BTs Hybrid Algorithm

During the movement to the target orbit, the UAVs approached the leader, forming a swarm (Fig. 6).

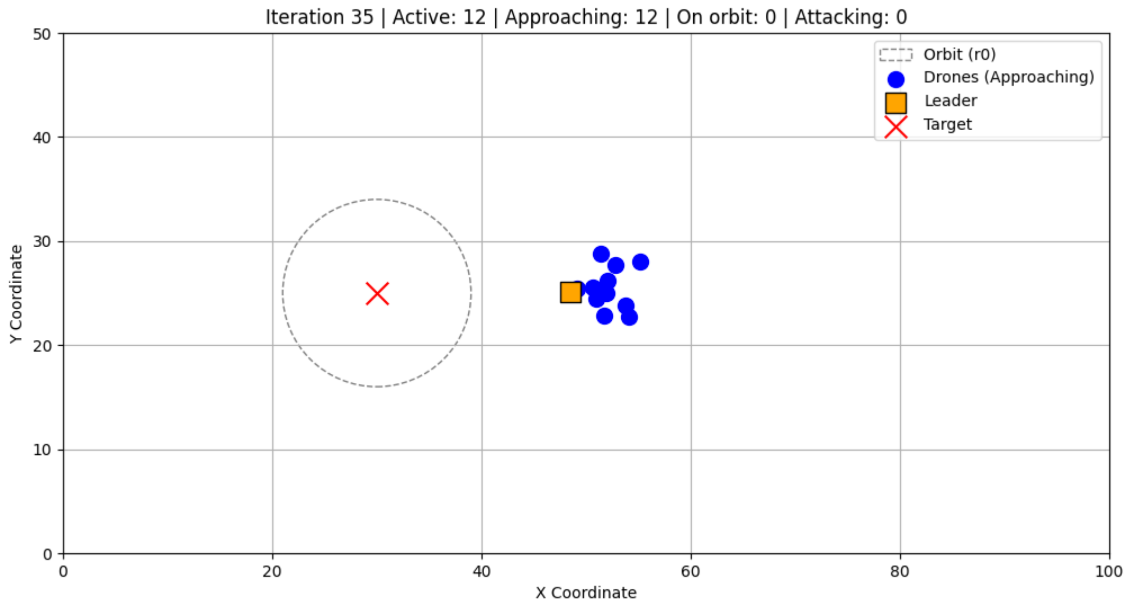

Figure 6: Movement of the UAV Swarm to the Target Orbit in a Simulation using the GBestPSO+BTs Hybrid Algorithm

Upon reaching the target orbit, the swarm distributed itself along the orbit (Fig. 7).

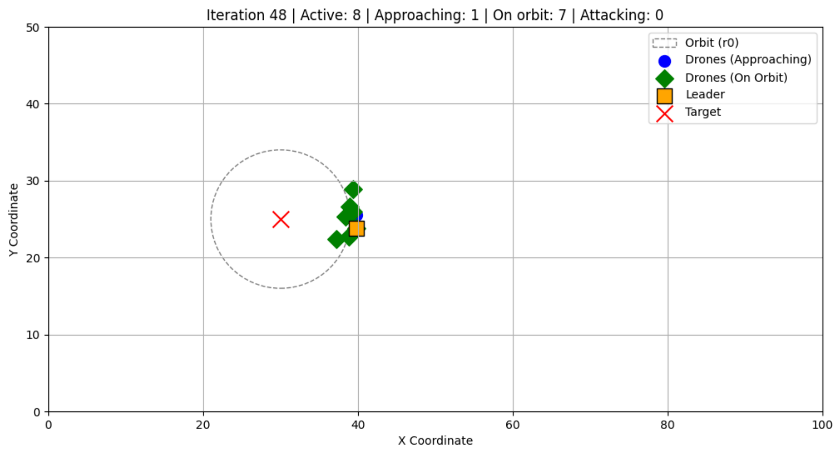

Figure 7: Distribution of the UAV Swarm along the Target Orbit

The swarm dispersed in orbit until the minimum number of UAVs required for the attack had gathered in orbit. After that, a simultaneous attack from different directions began (Fig. 8).

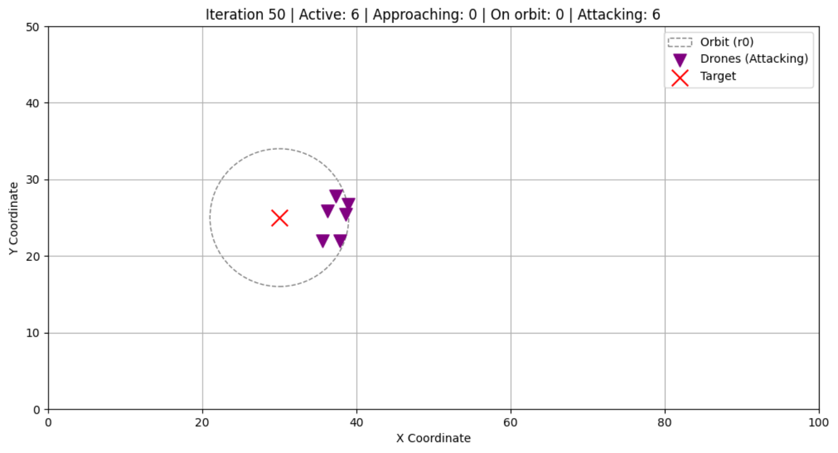

Figure 8: Simultaneous Attack on the Target with the Minimum Number of UAVs required for the Attack

The key metrics obtained during the experiments are shown in Table 5.

Table 5
Key Metrics for Simulating Scenario 1 using the GBestPSO+BTs Hybrid Algorithm

Metric	Value (average for 10 launches)
Average orbit formation time (iterations)	58.7
Average number of UAVs in orbit during attack	9
Average attack initiation time (iterations)	121
Successful attacks (UAVs that hit the target)	11
Average number of UAVs lost before attack	2.1
Overall percentage of successful simulations	100%

Figure 9 shows the evolution of the number of UAVs in different states (Approaching, Orbiting, Attacking) depending on iterations.

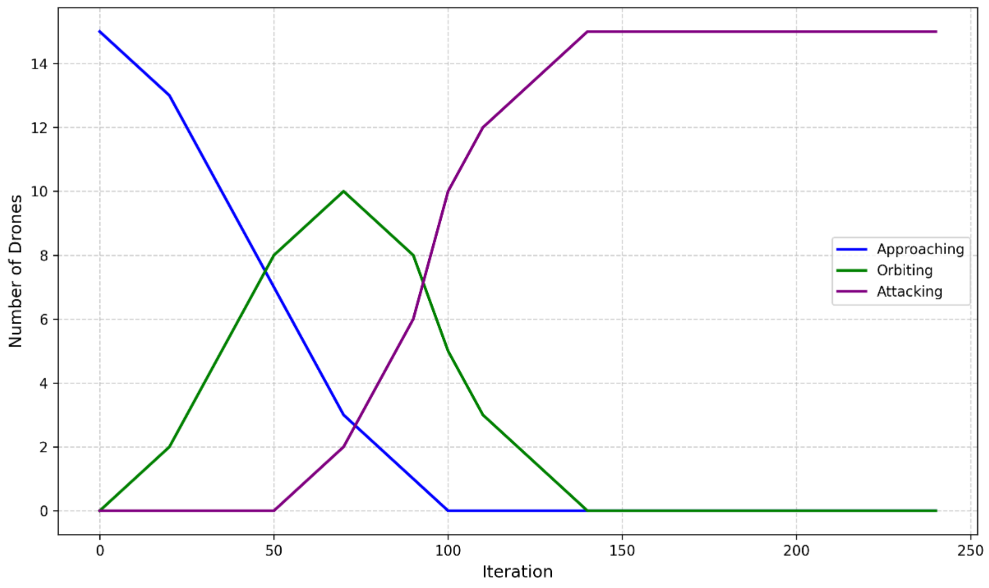

Figure 9: Dynamics of UAV states depending on iterations

Three distinct phases were observed:

- Phase I (0–50 iterations), during which most UAVs approach the target, with a few devices beginning to enter orbit.
- Phase II (50–120 iterations), when the number of UAVs in orbit steadily increases, reaching a uniform angular distribution.
- Phase III (after ~120 iterations), when a coordinated attack begins, leading to a sharp decrease in the number of UAVs in orbit and an increase in attackers.

The final series of experiments was conducted to simulate a swarm of UAVs for the “Reconnaissance of enemy territory” scenario. The swarm took off from the launch zone (Fig. 10) and was tasked with completely surveying a rectangular reconnaissance area located on the left side of the arena.

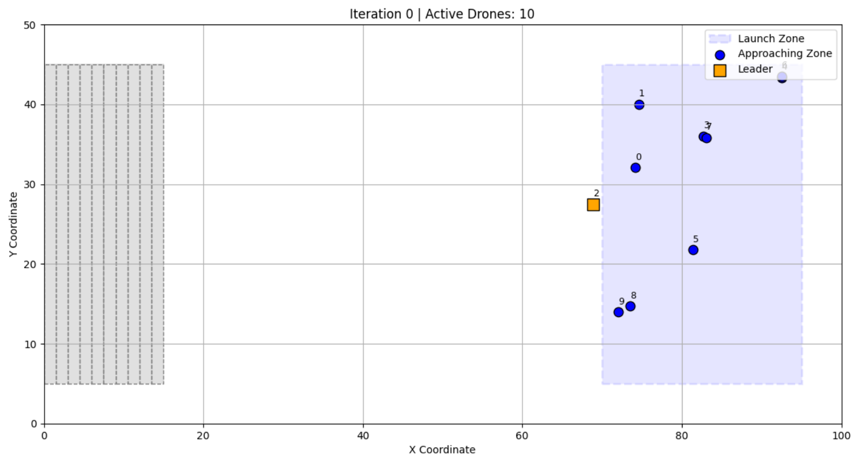

Figure 10: Start of the Mission for the “Reconnaissance of Enemy Territory” Scenario when simulated using the GBestPSO+BTs Hybrid Algorithm

The UAVs were assigned individual reconnaissance zones and coordinated their actions using behavior trees (BTs), while their movement and collision avoidance were controlled by the GBestPSO mechanism (Fig. 11).

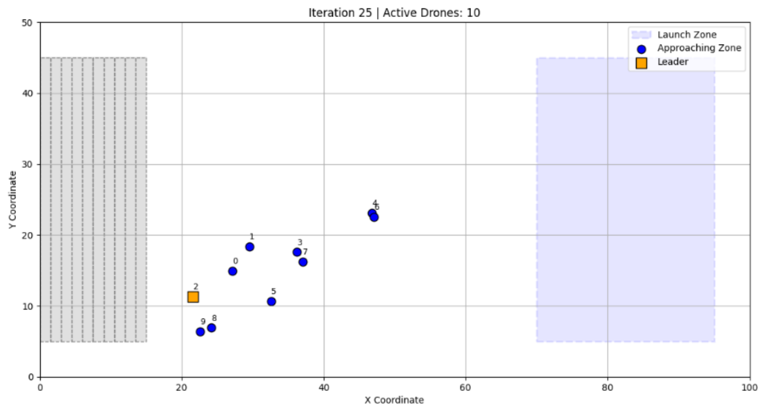

Figure 11: Movement of a Swarm of UAVs to a Reconnaissance Zone

Upon reaching the reconnaissance zone, each UAV began surveying a separate area (Fig. 12).

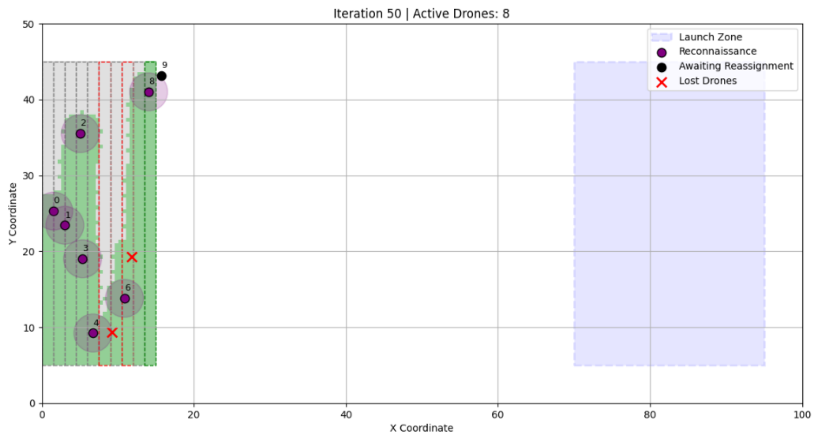

Figure 12: Surveying Separate Areas of the Reconnaissance Zone

After successfully surveying the entire reconnaissance area, the UAVs returned to their permanent deployment location (Fig. 13).

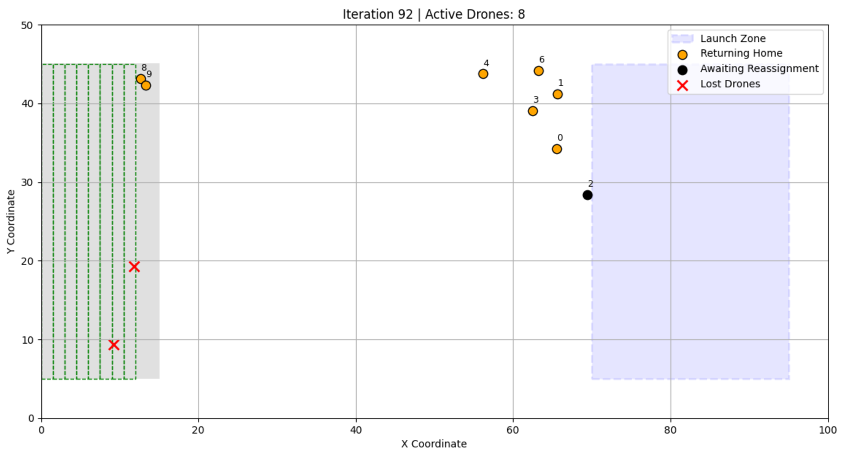

Figure 13: Return of UAVs to Their Permanent Deployment Location

For each simulation run, metrics were recorded that characterized the speed and quality of task execution, as well as the system's resistance to UAV losses.

Based on the research conducted, the following key metric values were obtained (Table 6).

Table 6
Key Metrics for Simulating Scenario 2 using the GBestPSO+BTs Hybrid Algorithm

Metric	Value (average for 10 runs)
Average time to reach the reconnaissance zone (iterations)	42
Average number of completed zones	9.6 out of 10
Average time to complete full coverage (iterations)	187
Average number of UAV losses	1.2
Percentage of completed missions	100%

The chart (Fig. 14) shows the dynamics of changes in the number of UAVs in different states over time.

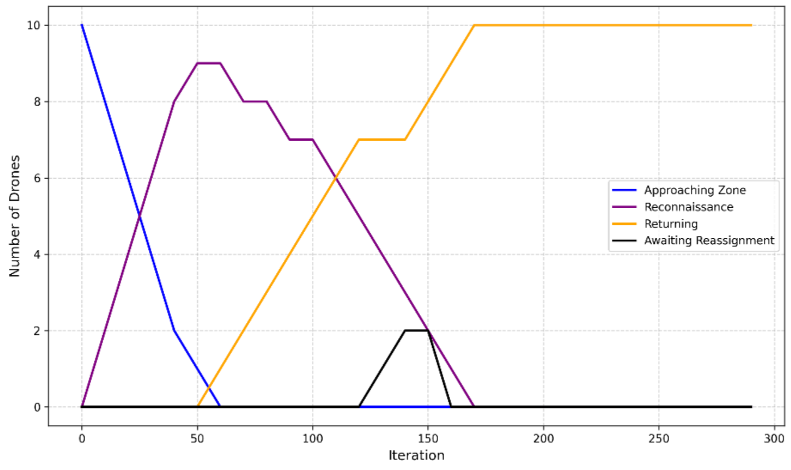

Figure 14: Dynamics of Changes in UAV States during Simulation

The dynamics of changes in the number of UAVs depended on the corresponding phase of the mission:

- Phase I (0–50 iterations) – most UAVs move to the areas; the first reconnaissance routes are gradually activated.
- Phase II (50–180 iterations) – the number of UAVs in Reconnaissance mode increases, gradually covering the entire area.
- Phase III (after ~180 iterations) – most UAVs complete their survey of the areas and switch to Returning or Awaiting mode.

6. Analysis and discussion

Experimental investigations confirm the effectiveness of the proposed hybrid approach, which combines swarm optimization (GBestPSO) and behavior trees (BTs). Analysis of the results for two key scenarios – "Swarm attack using the 'death ring' pattern" and "Reconnaissance of enemy territory" – allows us to draw reasonable conclusions about the advantages of this architecture.

A comparison of the results for the attack scenario demonstrates the clear advantage of the hybrid method over a system that uses only behavior trees. Although the BT-based approach provides a faster start to the attack (28–45 iterations versus 121 in the hybrid), it is inferior in terms of coordination quality. Agents controlled only by BTs move toward the target along the shortest trajectory, resulting in a less organized formation in orbit. In contrast, the hybrid system uses GBestPSO to pre-cluster the swarm around the leader, which, although it takes more time in the initial stage, provides significantly better synchronization and uniform angular distribution in orbit. This, in turn, leads to a more effective and simultaneous attack from different directions, minimizing the target's chances of countering.

In the reconnaissance scenario, the hybrid algorithm demonstrated high efficiency in solving space allocation and fault tolerance problems. The system successfully distributed the reconnaissance area between UAVs and ensured complete coverage of the territory, even under conditions of simulated agent losses. The two-level architecture plays a key role here: GBestPSO is responsible for strategic zone allocation and replanning in case of losses, while BTs control the tactical behavior of each UAV within its zone. This confirms that the proposed approach not only solves the problem of global optimization but also provides the local autonomy necessary to adapt to unpredictable circumstances.

Thus, the experiments prove the central thesis of the work: the integration of swarm intelligence for global planning and behavior trees for local execution creates a synergistic effect. GBestPSO provides the system with strategic coordination, while BTs provide tactical flexibility and responsiveness. This allows for a balance between global coordination and individual autonomy of agents, making hybrid architecture a promising solution for complex missions in dynamic environments.

7. Practical aspects and implementation

The proposed hybrid architecture is not only theoretically sound, but also practically implementable thanks to the use of modern technologies and modularity principles. The implementation is based on a two-level structure that includes planning (GBestPSO) and behavior (BT) levels, which interact through standardized messaging protocols.

The key advantages of the architecture in terms of implementation are modularity and scalability. Behavior trees are modular by nature, which makes it easy to reuse, extend, and modify the behavior logic of agents without having to redesign the entire system. Adding new UAVs to the swarm does not require changing the architecture, as GBestPSO works effectively with different numbers of agents, and each new agent functions as an independent unit with its own BT.

An important practical aspect is fault tolerance. The system is designed according to the principle of “weak connection” between levels. Even if communication with the global planner is lost, the agent can continue to perform tasks based on the last instructions received. The loss of individual UAVs also does not lead to mission failure, as the system is capable of redistributing tasks among active agents.

Finally, unlike resource-intensive methods such as deep reinforcement learning (DRL), the proposed approach is computationally efficient. This makes it suitable for implementation on UAV onboard computers with limited hardware resources, which is critical for practical application in the field.

8. Conclusions

The paper investigated and substantiated the effectiveness of a hybrid approach to coordinating autonomous agents, integrating swarm intelligence based on the GBestPSO algorithm and a control architecture based on behavior trees (BTs). The main problem addressed by the study is the need to ensure a balance between global coordination of group actions and local autonomy of each agent in dynamic and unpredictable environments.

The proposed two-level architecture, where GBestPSO is responsible for strategic planning and BTs for tactical reactive behavior, demonstrated high efficiency in simulation experiments. Two application scenarios were analyzed: coordinated swarm attack and territory reconnaissance.

Key results of the work:

- The hybrid approach was confirmed to be superior. In the attack scenario, the hybrid system, although requiring more time to prepare, provided a significantly higher level of coordination and synchronization of the strike compared to the approach based solely on BTs.
- Fault tolerance and adaptability were demonstrated. In the reconnaissance scenario, the system effectively distributed tasks among agents, adapted to UAV losses, and successfully completed the mission, confirming its reliability.
- The synergy of the two methods was substantiated. It was proven that the combination of global optimization using GBestPSO and local flexibility of BTs allows creating a system that is both purposeful and adaptive.

Thus, the main contribution of this work is to demonstrate that the integration of swarm intelligence and behavior trees is a promising direction for creating a new generation of multi-agent systems. Such systems are capable of functioning effectively in complex real-world conditions, making them suitable for a wide range of applications, from military operations to search and rescue missions.

9. Future research and applications

Despite the successful results, there are several promising areas for further development and improvement of the proposed hybrid approach. Future research may focus on the following aspects:

- Expanding the behavior tree library. Creating more complex and versatile sub-trees to implement a wider range of tactical actions, such as evading electronic warfare systems, dynamically changing roles (e.g., from reconnaissance to strike UAV), or cooperative interaction to perform complex tasks.
- Intellectualization of GBestPSO. Improving the swarm optimization algorithm by integrating mechanisms for adapting to dynamic changes in the environment in real time. For example, the algorithm parameters could be automatically adjusted depending on the threat level, obstacle density, or communication channel availability.
- Heterogeneous swarms. Adapting the architecture to manage heterogeneous swarms consisting of agents with different capabilities (e.g., UAVs for reconnaissance, electronic warfare, and strikes). This will require the development of more complex mechanisms for distributing tasks and roles at the GBestPSO level.
- Combination with learning methods. Investigation of the possibilities for synergy between the proposed approach and elements of machine learning. For example, reinforcement learning methods can be used for offline optimization of behavior tree parameters or GBestPSO, which will combine the advantages of transparency and verifiability of classical methods with the high efficiency of data-based models.
- Transition to physical testing. Conducting experiments using the "hardware-in-the-loop" model, followed by a transition to full-scale field tests on real UAVs. This will allow verifying the effectiveness of algorithms in conditions of real communication delays, sensor noise, and other physical limitations.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Falco, G. Robiolo, Tendencies in multi-agent systems: A systematic literature review, CLEI Electronic Journal 23 (2020). doi:10.19153/cleiej.23.1.1.

[2] A. Farinelli, A. Rogers, N. R. Jennings, Agent-based decentralised coordination for sensor networks using the max-sum algorithm, Autonomous Agents and Multi-Agent Systems 28 (2014). doi:10.1007/s10458-013-9225-1.

[3] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

[4] M. Colledanchise, P. Ogren, Behavior Trees in Robotics and AI: An Introduction, 2018. doi:10.1201/9780429489105.

[5] M. Iovino, E. Scukins, J. Styrud, P. Ogren, C. Smith, A survey of behavior trees in robotics and AI, Robotics and Autonomous Systems 154 (2022). doi:10.1016/j.robot.2022.104096.

[6] H. Karishma, A new hybrid particle swarm optimization algorithm for optimal tasks scheduling in distributed computing system, Intelligent Systems with Applications 18 (2023). doi:10.1016/j.iswa.2023.200219.

[7] Y. Shi, R. C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the IEEE Conference on Evolutionary Computation (ICEC), 1998, pp. 69-73. doi:10.1109/ICEC.1998.699146.

[8] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, 1999.

[9] M. Dorigo, M. Birattari, T. Stützle, Ant colony optimization, IEEE Computational Intelligence Magazine 1 (2006) 28-39. doi:10.1109/MCI.2006.329691.

[10] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, Journal of Global Optimization 39 (2007) 459-471. doi:10.1007/s10898-007-9149-x.

[11] F. Heylighen, Stigmergy as a universal coordination mechanism II: Varieties and evolution, Cognitive Systems Research 38 (2015). doi:10.1016/j.cogsys.2015.12.007.

[12] X.-S. Yang, Firefly algorithms for multimodal optimization, in: Lecture Notes in Computer Science, vol. 5792, 2010, pp. 169-178. doi:10.1007/978-3-642-04944-6_14.

[13] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC), 2009, pp. 210-214. doi:10.1109/NABIC.2009.5393690.

[14] N. Zainal, A. Zain, N. Radzi, A. Udin, Glowworm swarm optimization (GSO) algorithm for optimization problems: A state-of-the-art review, Applied Mechanics and Materials 421 (2013) 507-511. doi:10.4028/www.scientific.net/AMM.421.507.

[15] C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, in: Proceedings of SIGGRAPH '87, ACM Press, New York, NY, 1987, pp. 25-34. doi:10.1145/37401.37406.

[16] G. Xie, H. Xu, Y. Li, X. Hu, C.-D. Wang, Consensus enhancement for multi-agent systems with rotating-segmentation perception, Applied Intelligence 53 (2022). doi:10.1007/s10489-022-03687-x.

[17] M. Naserian, A. Ramazani, A. Khaki, A. Moarefianpour, Leader-follower consensus control for a nonlinear multi-agent robot system with input saturation and external disturbance, Systems Science & Control Engineering 9 (2021) 260-271. doi:10.1080/21642583.2021.1897959.

[18] C. Zhu, M. Dastani, S. Wang, A survey of multi-agent deep reinforcement learning with communication, Autonomous Agents and Multi-Agent Systems 38 (2024). doi:10.1007/s10458-023-09633-6.

[19] V. Mnih, et al., Human-level control through deep reinforcement learning, Nature 518 (2015) 529-533. doi:10.1038/nature14236.

[20] X. Jin, W. Zhengxiao, Proximal policy optimization based dynamic path planning algorithm for mobile robots, Electronics Letters 58 (2021). doi:10.1049/ell2.12342.
[21] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS '17), Curran Associates Inc., Red Hook, NY, 2017, pp. 6382-6393.
[22] J. Kober, J. Bagnell, J. Peters, Reinforcement learning in robotics: A survey, The International Journal of Robotics Research 32 (2013) 1238-1274. doi:10.1177/0278364913495721.
[23] A. Ulusoy, S. Smith, C. Belta, Optimal multi-robot path planning with LTL constraints: Guaranteeing correctness through synchronization, in: Springer Tracts in Advanced Robotics, vol. 104, 2012, pp. 447-462. doi:10.1007/978-3-642-55146-8_24.
[24] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, F. Yaman, SHOP2: An HTN planning system, Journal of Artificial Intelligence Research 20 (2003) 379-404. doi:10.1613/jair.1141.
[25] F. Medeiros, pyAutonomousAgent: An academic tool for modeling autonomous agent behaviors using behavior trees, Journal of Aerospace Technology and Management 16 (2024). doi:10.1590/jatm.v16.1352.