DOI: https://doi.org/10.15276/aait.09.2026.21
UDC 681.5:004.89

AI-assisted optimization of multi-agent system parameters: a model of iterative hyperparameter tuning in swarm intelligence algorithms

Mykyta S. Kozlov[1)]
ORCID: https://orcid.org/0009–0007–0071–7748; nikita.kozlov@stud.onu.edu.ua
Eugene V. Malakhov[1)]
ORCID: https://orcid.org/0000–0002–9314–6062; eugene.malakhov@onu.edu.ua. Scopus Author ID: 56905389000
Ravi Samikannu[2)]
ORCID: https://orcid.org/0000-0002-6945-6562; samikannur@biust.ac.bw
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, Vsevolod Zmiienko Str. Odesa, 65082, Ukraine
[2)] Botswana International University of Science and Technology, Palapye, Botswana

ABSTRACT

The relevance of the study is driven by the necessity for rapid and safe analysis of territories affected by emergencies, which stimulates the search for innovative approaches in the field of autonomous systems, in particular the use of swarms of unmanned aerial vehicles for terrain scanning. **The aim of the article** is to present an intelligent multi-agent superstructure for the dynamic adjustment of existing swarm system parameters, based on a combination of Swarm Chemistry self-organization mechanisms and global optimization using an evolutionary algorithm. **The task** consists in the development of an iterative tuning technology that allows for real-time adaptation of the behaviour of individual agents and the entire formation in response to dynamic environmental changes without interfering with the base structure of the control algorithm. **Among the methods** employed is an architecture consisting of a set of specialized agents organized into four functional layers, performing continuous monitoring, sensitivity analysis, and mission hyperparameter adjustment, together with a combined methodology of a genetic algorithm and local optimization methods that take into account swarm heterogeneity and the individual scanning widths of each device. **The scientific novelty** of the work lies in the dynamic tuning system architecture that ensures the survivability and high performance of existing swarm solutions in complex conditions. **The practical significance** is determined by the significant potential for application in disaster zone monitoring, search for victims, and real-time damage assessment. **The results** of simulation demonstrated a significant reduction in mission execution time and maximization of the coverage area while simultaneously increasing the stability of swarm behaviour, even under conditions of partial loss of agents or the appearance of unpredictable obstacles. **The conclusions** point to further development of the model, which involves the integration of machine learning methods for an in-depth analysis of previous mission experience and the expansion of the system for operation in three-dimensional space.

Keywords: Information technology; swarm intelligence; iterative tuning; multi–agent systems; hyperparameters; Swarm Chemistry; genetic algorithm; scan optimization

For citation: **Kozlov M. S., Malakhov E. V., Samikannu R., Shvets Y. O. "AI-assisted optimization of multi-agent system parameters: a model of iterative hyperparameter tuning in swarm intelligence algorithms".** ***Applied Aspects of Information Technology*****. 2026; Vol.9 No.3: 310–325. DOI: https://doi.org/10.15276/aait.09.2026.21**

INTRODUCTION

In modern conditions, society increasingly faces challenges caused by technological, natural, and anthropogenic emergencies. Large–scale fires, earthquakes, and infrastructure destruction resulting from military actions or accidents at critical infrastructure facilities require rapid response, which primarily depends on the timely acquisition of reliable information about the affected area.

However, involving humans in such operations often poses critical risks to life due to toxic emissions, radiation contamination, threats of landslides, or active combat operations.

This necessitates the need for autonomous aerial reconnaissance technologies capable of scanning terrain with minimal risk to rescuers.

One of the most promising directions is the use of swarm systems on unmanned aerial vehicles (UAVs). Multi–level drone complexes demonstrate a high level of adaptability, scalability, and fault tolerance due to decentralized control and flexible reconfiguration capabilities [1]. Modern experimental data confirm the possibility of optimized group flight (flocking) even in confined spaces and in the presence of dense obstacles [2]; however, this requires ideal tuning of interaction parameters.

The effectiveness of such systems largely depends on the internal configuration of the swarm movement algorithms, which directly determines:

– the degree of area coverage density;

– the uniformity of load distribution between nodes;

– the mission execution speed;

– the volume of overlaps or “blind zones.”

Traditional approaches treat parametric optimization as a static task solved before the mission begins. However, in dynamic conditions (changes in zone geometry, obstacles, wind load, etc.), pre-set hyperparameters quickly lose relevance. Incorrect configuration leads to duplication of effort, reduced observation quality, and excessive energy consumption. In complex scenarios, classical centralized methods or rule–based systems prove to be insufficiently flexible.

Modern approaches to swarm control include global optimization algorithms (PSO, ACO) and hybrid schemes with machine learning elements.

However, without proper refinement, these methods have significant limitations:

– low speed of real-time adaptation to environmental changes;

– excessive computational complexity for onboard systems;

– predominance of static planning strategies;

– inefficiency in the event of partial loss of agents or changes in their composition (heterogeneity).

To solve these problems, this paper proposes the implementation of an intelligent superstructure–a specialized multi–agent system (MAS) that functions in parallel with the main mission. Unlike single-level models, the proposed architecture utilizes the concept of “demon agents” for iterative parameter tuning in real time. This approach allows the tuning process to be transformed from a discrete (preliminary) task into a continuous autonomous cycle.

The use of multi-agent systems to implement iterative tuning is critical for ensuring:

1) functional separation of the process: isolating analytical tasks from the main flight control cycle guarantees system reactivity;

2) adaptation to dynamic uncertainty: the ability to adjust hyperparameters “on the fly” when the environmental context changes;

3) stability control: implementation of mechanisms for validating physical constraints and safe “rollback” to stable configurations;

4) effective operation of heterogeneous groups: individual tuning of each swarm node depending on its technical characteristics (scanning width, speed, scanning depth).

Consequently, this research focuses on the development of a comprehensive 'intelligent support' system designed to independently maintain the optimal operating regime of the swarm throughout the entire mission duration; such a system aims to minimize the impact of the human factor while simultaneously maximizing the efficiency and quality of data collection within hazardous emergency zones.

ANALYSIS OF LITERARY DATA

Modern methods for controlling swarm systems typically focus on trajectory planning or the formation of agent behavior rules within the boundaries of a specific mission. However, when it comes to complex hybrid architectures that combine self-organization based on “Swarm Chemistry” and global optimization via GA, the problem of static hyperparameters within the optimizers themselves arises.

Most existing analogues can be classified according to the specific method of system configuration adjustment:

1) static and evolutionary methods;

2) hybrid systems with reinforcement learning;

3) Reinforcement Learning (RL) hierarchical neuro-fuzzy networks.

Static and evolutionary tuning methods.

The first group of methods is characterized by a deterministic approach to parameter selection. The traditional use of GA [3] and particle swarm optimization (PSO) [4] involves searching for optimal solutions during the preliminary simulation stage. For instance, in models based on Swarm Chemistry, kinetic parameter recipes-such as the perception radius or cohesion and separation forces–are often calculated in advance and remain unchanged throughout the entire mission [5]. Additional studies of self-organization mechanisms emphasize that the complexity of local interactions within a swarm requires constant quantitative measurement to maintain structural stability [6].

Their advantage lies in high stability and low computational load during execution. However, the inability to adapt the optimization parameters themselves to sudden environmental changes (agent failure, changes in zone geometry) leads to the system getting stuck in local optima.

Consequently, their efficiency degrades whenever real-world conditions deviate from those assumed during the preliminary tuning stage.

Hybrid systems with RL

To address the limitations of static models, a second group of approaches utilizes RL components. These act as modern analogues attempting to overcome static limitations by implementing deep learning and strategy adaptations. Such an analogue, for example, is PSO-M3DDPG [7]. Such systems utilize neural networks for the dynamic updating of behavior policies in real time. However, they operate primarily at the level of drone movement control rather than at the level of adjusting the parameters of the optimization algorithms themselves. Furthermore, they possess excessive computational complexity, which complicates their integration as a superstructure without losing the performance of the core system.

RL hierarchical neuro-fuzzy networks

A third, more computationally efficient alternative to deep RL is the application of methods based on hierarchical neo-fuzzy networks (NF-RBFNN) [7]. This approach is based on using neo-fuzzy radial basis function neural networks to adapt system behaviour in dynamic scenarios. Unlike classical algorithms, such networks are capable of rapid learning based on input data streams, allowing the system to correct agent interaction strategies when external conditions change.

The Multi-Agent Superstructure for Hyperparameter Tuning (MAS-Tuning), unlike the considered analogues where optimization is an internal part of the movement logic, acts as an external intelligent layer over the existing tandem of evolutionary algorithms and “chemical” rules.

However, the main disadvantage of such systems is the difficulty of interpreting the decisions made (“black box”) and high sensitivity to the quality of the input sample. Moreover, implementing such a network as a superstructure requires significant computational resources for weight training in real time, which may negatively impact the stability of the primary scanning mission.

In summary, the analysis of existing approaches to meta-optimization allows us to highlight a series of systemic flaws that limit their application in real–world territory exploration missions using swarm complexes. The primary disadvantages are the static nature of most evolutionary strategies, which are unable to adapt “search rules”-to sudden environmental shifts, as well as the excessive computational complexity of hybrid neural network models, which leads to decision-making delays. Furthermore, the lack of mechanisms for intelligent validation and automatic rollback in traditional systems increases the risks of swarm destabilization during attempts at dynamic “on-the-fly” parameter tuning.

Collectively, these shortcomings motivate the need for a lightweight, interpretable, and failure-resilient optimization layer – one that can operate alongside the primary mission logic without compromising swarm cohesion.

PROBLEM STATEMENT

To date, within the field of unmanned systems management, there is an acute lack of universal instrumentation capable of performing flexible iterative adaptation of existing swarm algorithm parameters in accordance with the shifting local conditions of a mission. The unpredictability of terrain, the emergence of new threats, and the individual technical constraints of each separate drone require the system to move beyond simply executing a predefined program toward a state of continuous intellectual adaptation.

Addressing this scientific and technical challenge involves the implementation of a specialized multi-agent superstructure designed to ensure the rapid generation, validation, and application of potentially effective parameter configurations in real time. Such an approach will allow for a radical reduction in the total mission execution time and minimize undesirable duplication of scanning zones, all while preserving the integrity of the existing swarm control logic.

The intelligent superstructure under development must take into account the following critical aspects described below.

The first of these is the heterogeneity and individual characteristics of the agents. Since a swarm may consist of UAVs of various types, the system must recognize differences in their maximum movement speeds and specific sensor parameters–particularly the width of the field of view, which depends on the camera angle or the effective range of the sensors. This implies that iterative tuning must be differentiated: the optimizer agents must select configurations that maximize the potential of each individual drone, ensuring adaptive route construction without gaps in coverage or critical overlaps.

Next is the environmental dynamics and operational resilience. A scanning plan formed at the start of a mission inevitably encounters destabilizing factors during implementation: the appearance of new obstacles (buildings, meteorological phenomena [9], etc.), changes in the accessibility of certain sub–zones, or the sudden failure of one or several agents.

The multi-agent superstructure must maintain continuous dynamic reconfiguration of parameters based on up-to-date telemetry data. This requires the implementation of an "intelligent support" mechanism, where environmental changes are accounted for in the iteration of configuration recalculations to maintain the optimal mission tempo.

An equally important aspect is the formalization of efficiency metrics and target optimization. Within the scope of this study, efficiency is understood as the balanced minimization of a multi-objective cost function [10], which allows us to evaluate the fitness of each hyperparameter configuration.

This function is defined as follows:

$$F = \omega_1 \cdot \frac{T}{T_{max}} + \omega_2 \cdot \frac{D}{A_{total}} \to min, \qquad (1)$$

where T is the current mission execution time; T_{max} is the maximum allowable time (battery life constraint); D is the area of overlapping zones (duplication); A_{total} is the total area of the target territory; ω_1, ω_2 are weight coefficients that determine the priority of speed versus resource conservation ($\omega_1 + \omega_2 = 1$).

The optimization targets are categorized into two key indicators:

– scanning time (T): the total time interval required to reach the target percentage of area coverage (for example, 95 % or higher);

– overlap coefficient (D): the volume of area scanned multiple times by different drones, which indicates inefficient use of energy resources and duplication of efforts.

The final aspect is multi–level operational reconfiguration. The system must simultaneously support two types of reactions:

– local adaptation: instantaneous adjustment of an individual drone's parameters upon detecting an obstacle or local anomaly.

– global restructuring: a complete reconfiguration of the parameters for the entire swarm when the task topology changes, such as during the expansion of observation zone boundaries or a shift in sub-regional priorities.

To implement such an approach, the agent architecture must utilize decentralized coordination and a constantly updated map of current coverage, allowing efforts to be focused on areas of the territory that have not yet been processed.

Thus, the problem statement consists of the integration of an agent layer that would act as an "intelligent regulator", continuously adjusting the parameters of the existing system to the realities of the flight task.

RESEARCH GOAL

The objective of this research is to enhance the operational efficiency of existing swarm systems through the development and implementation of an adaptive multi-agent information technology for iterative tuning. The proposed approach is based on the integration of an intelligent superstructure consisting of 12 specialized agents. The core focus of this system is the minimization of the scanning time interval required to reach the target coverage. This objective is achieved alongside the maximization of the territory coverage area through the dynamic, real-time adjustment of hyperparameters within the base algorithm of the swarm complexes [1]. Technology ensures high precision by carefully accounting for the inherent heterogeneity of the swarm composition and the volatility of environmental conditions, such as unpredictable obstacles or agent failures, encountered during mission execution.

To achieve this stated objective, it is necessary to resolve the following specific research tasks:

1) to analyze existing approaches to parametric optimization within swarm complexes and to identify the fundamental limitations associated with static configurations;

2) to design the architecture of the intelligent superstructure, ensuring its seamless integration into the existing control loop through asynchronous messaging mechanisms;

3) to develop a methodology for the iterative search of optimal parameters that utilizes a combination of a ga for solution space exploration and various local optimization methods;

4) to formulate a comprehensive system of constraints and validation protocols that guarantee the preservation of the base algorithm's stability during dynamic parameter shifts;

5) to investigate mechanisms for automatic rollback to stable states in the event that degradation of the target efficiency metrics is detected.

Consequently, the result of this study is not merely a new control algorithm, but rather a holistic optimization superstructure model which:

– adapts to environmental shifts and drone performance characteristics by dynamically correcting the parameters of the existing algorithm without interfering with its fundamental underlying structure;

– maximizes the total coverage zone and minimizes the overall mission execution time by identifying the most effective operating regimes for any given situational context;

– operates efficiently during swarm scaling by distributing the computational workload among independent agents;

– allows for the flexible reconfiguration of the system in real-time mode, thereby ensuring a continuous process of system self-improvement without a loss in its overall productivity.

ARCHITECTURE DESCRIPTION

The proposed architecture is founded upon the principle of asynchronous iterative tuning, which enables the intelligent superstructure to function as an independent analytical hub positioned above the existing swarm control loop. The core idea lies in the creation of a closed-loop feedback cycle that operates in parallel with the Swarm Chemistry and genetic search algorithms described in the baseline system [10], [14], [15]. Such an approach allows for a radical separation between the "physical" layer, responsible for the direct movement of drones and scanning operations, and the "cognitive" layer, which specializes in continuous meta-optimization. Instead of interfering with every single step of trajectory calculation, the superstructure acts through a "soft update" mechanism for configurations, providing the existing system with enhanced sets of hyperparameters whenever they are ready, without halting the execution of the primary mission.

The selection of an architecture consisting specifically of 12 specialized agents is necessitated by the requirement to ensure complete functional isolation and high system survivability under real-time conditions (Fig. 1).

The number of agents was determined based on the division of the optimization cycle into four functional layers, with three agents operating within each layer. This "3×4" distribution is not arbitrary: it guarantees that at every stage – from data perception to execution – there exists an internal division of roles between information collection, critical verification, and decision-making. For instance, having three agents in the tuning layer allows for the simultaneous conduct of global search (Exploration), local refinement (Exploitation), and rigorous validation of constraints (Constraint Satisfaction). A smaller number of agents would lead to the overloading of individual modules and the emergence of computational "bottlenecks", whereas a larger number would significantly complicate inter–process interaction and increase the overhead costs associated with messaging.

MONITORING LAYER

The Monitoring Layer functions as the fundamental receptor system of the entire superstructure, providing "intelligent meta–vision" over the active swarm mission. The primary scientific and technical justification for isolating this layer into a distinct triad of agents lies in the necessity for complete separation of data collection and primary processing tasks from the core computational logic of Swarm Chemistry. Such a configuration ensures that the UAV hardware resources are expended exclusively on executing kinetic movement algorithms, while the superstructure operates with a "digital twin" of the mission asynchronously, thereby avoiding any delays in critical control cycles.

Main part of this information hierarchy is the Telemetry Collector agent from Monitoring layer, which serves as the single source of truth for all subsequent tuning stages. Rather than polling each UAV directly, the Telemetry Collector receives pre-aggregated telemetry packets from the swarm's queen node, which serves as the in-swarm aggregation point and forwards consolidated metrics – spatial coordinates, movement vectors, and energy resource status – at a rate of 10 times per second. The queen transmits data only for drones that successfully reported within the current sampling window; non-responding agents (signal loss, occlusion, and partial failure) are omitted from the packet rather than blocking transmission. This shields the superstructure from direct dependence on per-drone uplinks and bounds inbound bandwidth at the superstructure interface, since the TelemetryCollector handles a single aggregated stream from the queen regardless of swarm size. This sampling rate was deliberately chosen to strike a balance between data freshness and system load: collecting metrics more frequently would introduce unnecessary computational overhead, while a lower rate would leave optimization modules operating on a stale swarm model, effectively neutralizing the benefits of parameter correction. The unified message queue formed by this agent allows the system to avoid direct queries to physical devices.

Operating in parallel with data accumulation is the HealthMonitor agent, whose role consists not of passive collection, but of active operational stability assurance through a stagnation detection mechanism. It analyzes the growth rates of area coverage and identifies moments when the current hyperparameter configuration ceases to yield

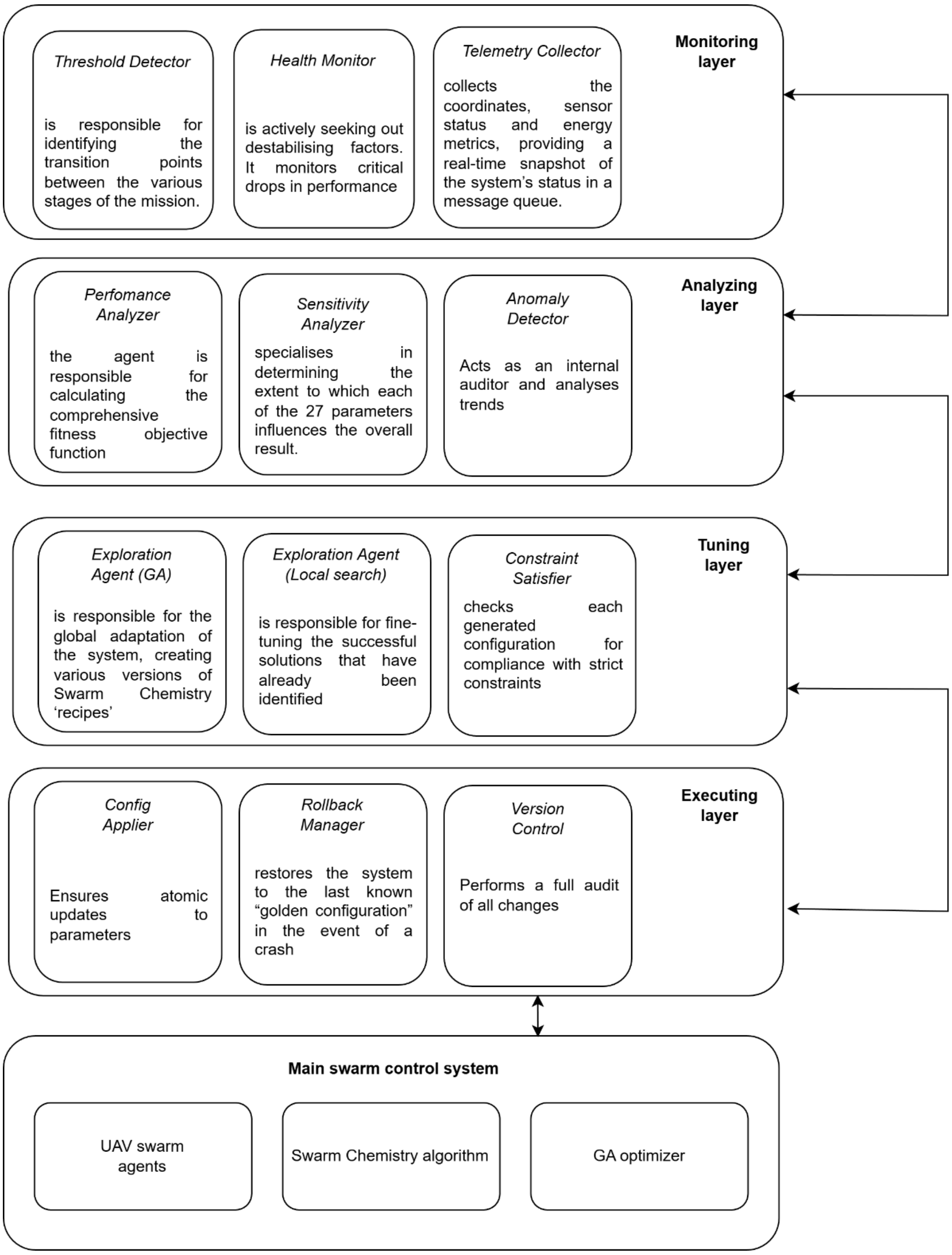


Fig. 1. **Multi-agent system architectural View**
***Source*: compiled by the authors**

results–for instance, due to drones looping in local minima. its verdict acts as an unconditional trigger for the immediate activation of the tuning layer. This activity is complemented by the ThresholdDetector, which is responsible for strategic phase adaptation. Since a scanning mission progresses through stages from initial deployment to the cleanup of minor gaps, this agent tracks critical execution thresholds and signals the system whenever a shift in optimization priorities is required, allowing the superstructure to remain context-dependent at every stage of the mission.

COGNITIVE EVALUATION AND META-ANALYTICS OF PARAMETERS

The Analysis Layer serves as the cognitive core of the technology, where the transformation of structured telemetry into the knowledge necessary for decision-making regarding hyper parameter changes occurs. Within the framework of iterative tuning, this layer acts as a critical connective link, as it provides the essential feedback for evolutionary mechanisms, mathematically justifying the expediency of each potential configuration.

The key function of performance evaluation is performed by the Performance Analyzer, which calculates a multi–factor fitness function (1). Unlike basic systems, this agent does not merely record the coverage percentage; instead, it evaluates the dynamic balance between the scanning speed and the energy expenditures of a heterogeneous swarm. The justification for such an approach lies in the necessity of accounting for the individual constraints of each drone: for units with a critical battery level, the agent calculates a higher “cost” for every meter of the trajectory, forcing the tuning system to search for configurations with minimal path duplication.

To accelerate the search process within an exceptionally vast parameter space ($>10^{20}$) the Sensitivity Analyzer is employed. This agent conducts sensitivity analysis, isolating from the 27 hyperparameters of the swarm control system (see Table 1 and the description of the augmented hyperparameter vector below) those that exert a dominant influence on the current situation (for instance, the priority of unscanned zones during the initial stages of a mission). This allows the computational resources of the GA to be focused exclusively on significant constants, radically reducing the optimization convergence time.

ITERATIVE SEARCH AND VERIFICATION OF OPTIMAL CONFIGURATIONS

The Tuning Layer serves as the operational core of the intelligent superstructure, where the direct generation of new hyperparameter sets for the existing system takes place. The scientific novelty of the approach at this stage lies in the transition from static evolutionary search to a multi-level strategy of “adaptive adjustment” in real-time mode. The allocation of three specialized agents within this layer allows for the simultaneous resolution of conflicting tasks: the necessity for global exploration of new states (exploration) and the requirement for the fastest possible local improvement of current efficiency (exploitation), without the risk of violating the swarm's physical constraints.

The primary role in the global adaptation process is played by the Exploration Agent, which is based on a modified GA. Utilizing the fitness evaluation results from the analysis layer, it operates on a population of configurations, applying crossover and mutation operators to search for radically new combinations Swarm Chemistry hyperparameters [10], [20]. Its operation provides the system with the ability to escape local minima, which is critical when the topology of the scanning zone changes or when a portion of the agents is lost. In contrast, the ExploitationAgent focuses on local search methods, such as the hill climbing algorithm [11]. It performs micro–corrections of the most influential parameters identified by the sensitivity analyzer, allowing the system to achieve peak scanning efficiency [12], [13] within the current region of the state space significantly faster than a classical GA would.

The triad of this layer is completed by the ConstraintSatisfier agent, which functions as an intelligent safety filter. Any configuration generated by the preceding agents undergoes mandatory verification for compliance with a system of "hard constraints." This is justified by the need to preserve the integrity of the base algorithm: for instance, the agent prevents the setting of repulsion or attraction force values that would lead to physically impossible accelerations or the violation of safe distances between UAVs. Configurations that fail validation are not simply discarded – instead, the agent attempts to project them onto the nearest permissible point within the constraint space, salvaging promising parameter directions while correcting only the violating dimensions. Thus, the tuning layer outputs only verified and potentially effective parameter sets, ready for immediate implementation.

SAFE IMPLEMENTATION AND SURVIVABILITY MANAGEMENT

The Execution Layer facilitates the final stage of the iterative tuning lifecycle – the secure transition of the existing system to a new hyperparameter configuration. The primary objective of this level is to guarantee the atomicity of updates and maintain the high survivability of the technology in cases where theoretically optimal parameters might lead to an unexpected degradation of actual mission performance metrics. The utilization of three specialized agents within this layer allows for the clear separation of the change application process from risk insurance mechanisms and the accumulation of historical experience.

Direct interaction with the software interface of the existing system is managed by the ConfigApplier. Its core function lies in the synchronous updating of Swarm Chemistry parameters for all active swarm agents. The justification for the asynchronicity of the entire superstructure is most clearly demonstrated here: the drones continue their movement based on previous stable settings until the ConfigApplier receives a signal regarding the readiness of a new, verified configuration. The replacement itself occurs instantaneously, which eliminates any state of system uncertainty where one part of the swarm might operate under new rules while the other remains under the old ones.

The RollbackManager ensures overall reliability by continuously comparing the predicted efficiency of each new configuration against actual telemetry from the monitoring layer. If a drop in coverage metrics or a critical increase in energy consumption is detected within a short time window, the agent immediately reverts to the last "golden" (guaranteed stable) configuration, allowing the system to conduct bold optimization experiments without the threat of mission failure. In parallel, VersionControl archives every successful tuning iteration along with its application context (terrain type, number of drones), forming a knowledge base that enables progressively faster parameter initialization under similar future conditions – transforming the superstructure from a reactive tool into a proactive intelligent system.

The end-to-end data flow of the iterative parameter tuning workflow is summarised in Fig. 2. The diagram traces a single optimisation cycle from telemetry ingestion through sensitivity-driven population update, constraint verification, deployment, and the rollback path that activates upon performance regression.

ALGORITHMIC SUPPORT FOR ITERATIVE PARAMETER TUNING

The functioning of the intelligent superstructure is based on the synergy of two methodologies: global genetic search for state space exploration and local gradient–based methods for refining the discovered configurations. The primary complexity of the algorithmization lies in the necessity of ensuring the convergence of the optimization process under tight time constraints, where the objective function (scanning efficiency) is constantly shifting due to the dynamics of the real–world environment.

At the core of the tuning layer's operation is an adaptive evolutionary algorithm, where each "individual" in the population represents a 27-dimensional hyperparameter vector. This vector consolidates the 22 base parameters of the chaotic-deterministic swarm control method (introduced in [10], [20] and grouped into four functional categories: social interaction forces, attractiveness function, anti-stagnation mechanisms, and trajectory conditioning) together with 5 additional formation-specific parameters introduced in the present work to support the dual-ring orbital structure of the Mixed Formation Movement mode**.** These parameters modulate the underlying dynamics of the existing Swarm Chemistry system [10], [20], directly influencing the kinetic interactions and emergent behavior of the swarm rather than imposing rigid control constraints. The composition of the additional formation-specific parameters is summarised in Table 1.

The process begins with the initialization of the population based on the current stable configuration. The Exploration Agent applies crossover operators to combine successful traits from different configurations and a mutation operator to prevent premature convergence into a local optimum. The scientific justification for using an evolutionary approach specifically lies in its ability to effectively process multidimensional, non–linear, and noisy data, such as UAV swarm telemetry.

In parallel with the global search, the Exploitation Agent implements a "greedy" refinement mechanism. Since GA excel at finding promising regions but may converge slowly toward an exact maximum, the local search focuses on the most sensitive parameters identified by the analysis layer. This allows the system to perform minute correction steps (for instance, adjusting the attraction force toward unscanned zones by 1-2 %), yielding an immediate performance boost without waiting for

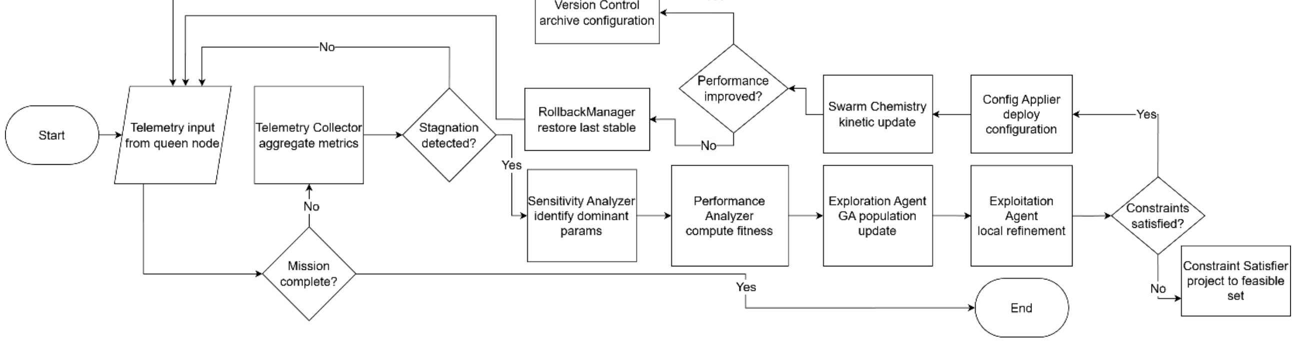


Fig. 2. **Iterative parameter tuning workflow**
Source: **compiled by the authors**

Table 1. **Formation-specific hyperparameters**

Parameter	Range	Description
Formation_radius_outer	[40, 150]	Radius of the outer ring of the formation; outer-tier drones orbit at this distance from the formation centre.
Formation_radius_inner	[15, 60]	Radius of the inner ring of the formation; inner-tier drones orbit at this distance from the centre.
Orbit_speed	[0.01, 0.30]	Angular velocity of orbital rotation, expressed in radians per simulation timestep. Higher values produce faster rotation and denser tangential coverage; lower values yield slower, more deliberate sweeps.
Random_drift_strength	[0.00, 0.50]	Amplitude of Gaussian noise superimposed on the orbital motion. A value of zero corresponds to deterministic orbital trajectories; higher values introduce stochastic perturbations that prevent repeated coverage of identical paths.
Outer_drone_fraction	[0.15, 0.85]	Fraction of swarm agents assigned to the outer ring; the remainder are placed on the inner ring. Larger values yield wider, sparser coverage patterns; smaller values produce a denser, more compact formation.

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

the completion of a full population evolution cycle. Such a combination enables the superstructure to be both strategically flexible and tactically precise.

The mathematical support for iterative tuning requires a clear formalization of success criteria. The objective function (f_{obj}), calculated by the Performance Analyzer agent, is an additive weighted aggregation of several metrics: maximizing the area coverage percentage (C), minimizing the elapsed time (T) and minimizing the duplication coefficient (D):

$$f_{obj} = \omega_C \cdot \left(1 - \frac{C}{100}\right) + \omega_T \cdot \frac{T}{T_{max}} + \\ +\omega_D \cdot \frac{D}{A_{total}} \rightarrow min, \quad (2)$$

where C is the current area coverage percentage (subtracted from 1 to convert maximization into a minimization task); T is the elapsed mission time; D is the area of overlapping zones (duplication); $\omega_C, \omega_T, \omega_D$ are dynamic weight coefficients for coverage, time, and duplication respectively.

Crucially, the weight coefficients of these metrics are not static; they are dynamically adjusted by the Threshold Detector depending on the current mission phase (ϕ).

This adaptation can be represented as a function of the mission progress:

$$W(\phi) = \{\omega_C(\phi), \omega_T(\phi), \omega_D(\phi)\} | \sum \omega_i = 1. \quad (3)$$

For example, the adjustment of the duplication weight (ω_D) follows a non-linear growth pattern as the mission approaches completion:

$$\omega_D(\phi) \ = \ \omega_{D_0} + \alpha \cdot e^{\beta \cdot \frac{C}{100}}, \quad (4)$$

where ω_{D_0} is the initial weight, while α and β are scaling factors that amplify the importance of avoiding overlaps during the final stage of scanning. This encourages the system to find parameters that direct drones into narrow unscanned gaps while avoiding already processed areas, ensuring a high-precision “clean-up” phase.

A mandatory element of the algorithm is the Constraint Satisfaction block. Since Swarm Chemistry is based on a balance of forces (attraction, repulsion, alignment), a random combination of parameters could lead to the “disintegration” of the swarm or its collapse into a single point. The validation algorithm imposes a rigid system of inequalities on the generated parameter vectors. If a proposed configuration violates even a single physical constraint (for example, $R_{repulsion} > R_{attraction}$), it is either automatically rejected or adjusted to the nearest permissible value. This guarantees that iterative tuning always remains within the stability boundaries of the existing system, ensuring its reliable operation in real–world conditions.

EXPERIMENTAL SETUP AND RESULTS

To empirically evaluate the proposed 12-agent MAS tuning superstructure, a controlled simulation study was conducted using the project's chaotic area movement system. All experimental runs were orchestrated via the Claude Code API (Anthropic) [16], [17], which automated batch simulation management, statistical analysis, and visualization generation. This integration enabled the full experimental cycle – from simulation launch to result aggregation – to be executed and reproduced without manual intervention, ensuring methodological consistency. Results reported in this section are based on 100 runs, providing a sufficient statistical foundation for the conclusions drawn.

Simulation Environment and Configuration

The experimental environment was modelled as a 53×40 grid (2120 cells) representing a flat aerial scanning zone. Table 2 summarizes the fixed configuration parameters shared by both experimental conditions.

Table 2. **Fixed Simulation Parameters (Both Conditions)**

Parameter	Value
Grid size	53×40 cells (2120 total)
Drone class	Matrice
Initial energy	593 Wh per drone
Static obstacles	3 per run
Obstacle density	6.6%–22.9% (run-dependent)
Time limit	60 s (hard cutoff)
Movement mode	Global chaotic (adaptive params ON)
Rendering	Headless (no GUI overhead)
Runs per	50 independent runs

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

Two experimental conditions were compared: Condition A (no-MAS) – the standard chaotic algorithm with territory-based adaptive parameter control but without the MAS superstructure; and Condition B (with MAS) – the identical base system augmented by the full 12-agent MAS architecture, with all four functional layers (Monitoring, Analysis, Tuning, Execution) active throughout the mission. The chaotic algorithm's hyperparameter seed was identical at the start of both conditions, sourced from the best configuration produced by the GA optimization phase described below.

Role of the Genetic Algorithm as a Pre-Stabilization Stage

A critical prerequisite for obtaining interpretable results under the MAS condition was the application of the GA [10] as a pre-experimental stabilization stage. Without this step, the Exploration Agent's unconstrained proposals led to configurations with values of unscanned area bias and swarm cohesion strength that destabilize collective dynamics, reducing final coverage below 10 % in the majority of runs.

The GA's role is formally analogous to a trust-region method in continuous optimisation: rather than acting as an online optimiser, it defines the parameter subspace Θ_{safe} within which the MAS agents propose modifications. Letting Θ^*_{GA} denote the best GA-evolved configuration, the constraint imposed on MAS proposals during a mission of duration T is:

$$\| \Theta_{MAS}(t) - \Theta^*_{GA} \|_\infty \leq \delta_{max}, \quad \forall t \in [0, T], \quad (5)$$

where δ_{max} is the per-parameter maximum deviation enforced by the ConstraintSatisfier agent. In the present experiments, this bound was implicitly set by the ConstraintSatisfier's physical-limit validation layer, effectively restricting the search to the neighbourhood of Θ^*_{GA}.

Performance Objective and Fitness Function

The Performance Analyzer agent computes a multi-factor fitness function used by the Tuning layer to evaluate candidate configurations. The objective combines three mission metrics in an additive convolution with dynamically adjusted weights:

$$F(\theta, t) = w_C(t)C(t) - w_T(t)\frac{t}{T} - w_D(t)D(t), \quad (6)$$

where $C(t) \in [0,1]$ is the fraction of the scan zone covered at time $t, D(t) \in [0,1]$ is the redundancy coefficient (fraction of cells scanned more than once), and T is the mission time limit. The weighting vector $w(t) = (w_C(t), w_T(t), w_D(t))$ is adjusted by the ThresholdDetector agent as the mission progresses through phases: during the exploration phase ($C < 0.5$) the weight wC dominates; during the completion phase ($C > 0.7$) the weight w_D increases to redirect drones toward unscanned gaps rather than revisiting covered cells.

Statistical Methodology

Prior to hypothesis testing, the normality of each metric distribution was assessed using the Shapiro-Wilk test. The test rejected normality for final coverage in both group (no-MAS: $W = 0.931, p = 0.006$; MAS : $W = 0.667, p < 0.001$) and for total distance traveled in the MAS group ($W = 0.570, p < 0.001$), consistent with the presence of three zero-valued failure runs in the MAS condition. Given the violation of normality assumptions, the Mann-Whitney U test was selected as the primary inferential tool:

$$U = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} 1[X_i - Y_j], \quad (7)$$

where X_i denotes observations from Condition A (no-MAS) and Y_j from Condition B (MAS), with $n_1 = n_2 = 50$. Effect size was quantified by the rank-biserial correlation:

$$r = 1 - \frac{2U}{n_1 n_2}. \quad (8)$$

Effect magnitudes were interpreted as negligible ($|r| < 0.10$), small ($|r| < 0.30$), medium ($|r| < 0.50$), and large ($|r| \geq 0.50$). All tests were two-sided with $\alpha = 0.05$. Result stability was quantified by the coefficient of variation (CV):

$$CV = \frac{\sigma}{\mu} \times 100\% \,. \quad (9)$$

Table 3 presents the complete descriptive statistics for all six primary metrics across both conditions. The CV serves as the principal stability indicator.

Primary Results: Coverage Performance

The primary outcome measure, final coverage percentage, revealed a statistically significant advantage for the MAS condition. As shown in Fig. 3, the MAS group achieved a mean coverage of $38.30\% \pm 15.04\,\%$ (median $42.68\,\%$), compared to $37.57\,\% \pm 5.04\,\%$ (median $37.79\,\%$) for the no-MAS baseline.

The same directional advantage was observed for average region coverage (mean $54.78\,\% \pm 21.60\,\%$ for MAS vs. $54.01\,\% \pm 7.65\,\%$ for the baseline). The six-panel overview in Fig. 3 presents the complete distributional comparison across all primary metrics, including region coverage and total distance traveled.

Mann-Whitney U Test Results

Table 4 presents the full Mann–Whitney U test [18], [19] results for all six primary metrics. All metrics reach statistical significance at $\alpha = 0.05$ and $\alpha = 0.01$, with medium-to-large effect sizes. For coverage, the Mann-Whitney test yielded $U = 705.0, p = 1.74 \times 10^{-4}$, with effect size $r = 0.436$ (medium). Region coverage confirmed this pattern ($U = 712.0,\ p = 2.11 \times 10^{-4}, r = 0.430,\ medium$). The direction of both effects was MAS > no-MAS. Simulation time produced a perfect separation ($U = 0.0, r = 1.000$), reflecting the deterministic nature of MAS thread initialization overhead. The negative effect sizes for distance ($r = -0.387$) and energy ($r = -0.387$) indicate lower values under MAS; however, as discussed earlier, this is primarily a consequence of three complete zero-output failure runs rather than a genuine improvement in path efficiency.

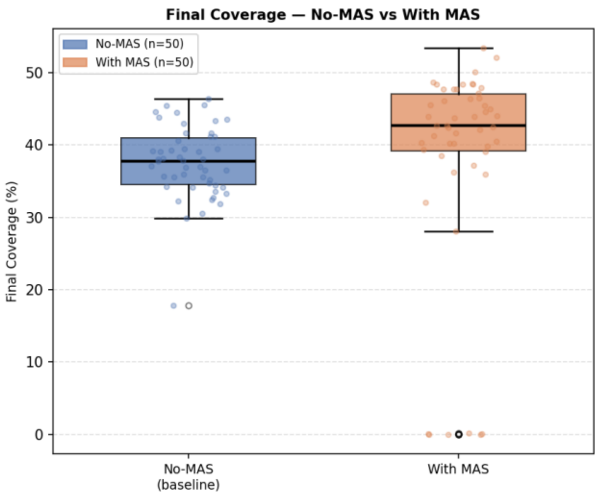


Fig. 3. **Final coverage percentage distribution**
***Source*: compiled by the authors**

The most consequential finding of this study concerns outcome stability. Fig. 4 presents the CV for all six metrics side by side, revealing a consistent and dramatic increase in variability under the MAS condition. The CV for final coverage under MAS is $39.3\,\%$, compared to $13.4\,\%$ for the no-MAS baseline – an $2.9 \times$ increase. The same ratio holds for all other metrics: distance and energy CV rise

Table 3. **Descriptive statistics for all primary metrics**

Metric	Condition A – No-MAS (baseline)				Condition B – With MAS			
	Mean	Std	Median	CV (%)	Mean	Std	Median	CV (%)
Final Coverage (%)	37.57	5.04	37.79	13.4	38.30	15.04	42.68	39.3
Avg Region Coverage (%)	54.01	7.65	55.20	14.2	54.78	21.60	61.25	39.4
Redundant Scans (%)	33.13	4.36	32.74	13.2	33.66	13.05	38.11	38.8
Simulation Time (s)	60.76	0.05	60.77	0.1	64.43	1.92	65.75	3.0
Distance Traveled (grid units)	24.438	1.472	24.383	6.0	21.040	7.517	23.074	35.7
Energy Consumed (Wh)	586.1	35.3	584.8	6.0	504.7	180.1	553.3	35.7

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

Table 4. **Mann-Whitney U Test Results**

Metric	U	p-value	r	Effect
Final Coverage (%)	705.0	1.74×10^{-4}	+0.436	Medium
Avg Region Coverage (%)	712.0	2.11×10^{-4}	+0.430	Medium
Redundant Scans (%)	705.0	1.74×10^{-4}	+0.436	Medium
Sim. Time (s)	0.0	7.07×10^{-18}	+1.000	Large
Distance (grid u.)	1734.0	8.59×10^{-4}	−0.387	Medium
Energy (Wh)	1734.0	8.59×10^{-4}	−0.387	Medium

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

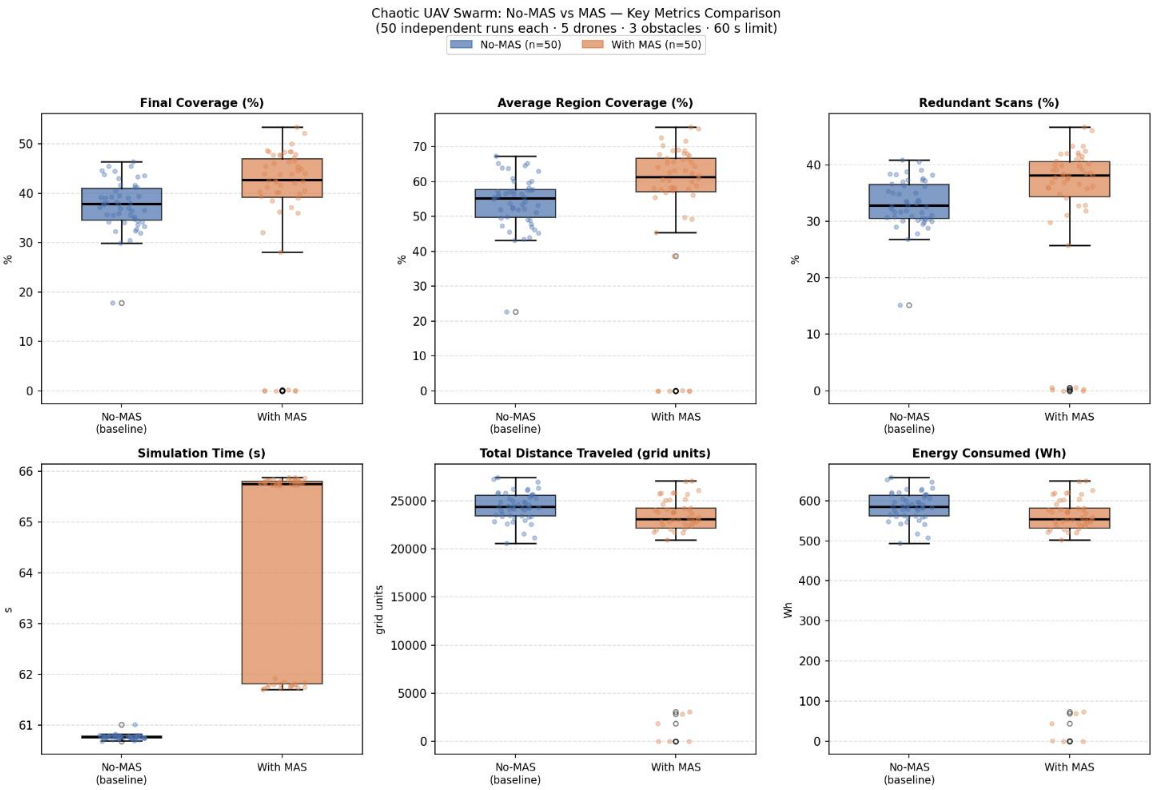


Fig. 4. **Six-panel box-plot comparison of all primary mission metrics for Condition A (no-MAS, blue) and Condition B (with MAS, orange)**
***Source*: compiled by the authors**

from ≈ 6 % to ≈ 36 %. This pattern confirms that while the MAS superstructure shifts the central tendency of coverage upward, it simultaneously introduces a level of outcome unpredictability that is not present in the unaugmented system.

Post-hoc investigation identified three complete-failure runs in the MAS condition (run indices 0, 33, and 49; failure rate 6.0 %), in which final coverage, total distance, and energy consumed were all recorded as zero despite the full 60-second wall-clock time having elapsed.

The failure pattern is consistent with a threading race condition during MAS start-up: the 12 agent threads and fanout relay threads are launched concurrently with the simulation loop, and in a subset of runs the queue initialisation sequence blocks drone activation for the entire mission duration. All three failure runs were included in the primary statistical analyses to faithfully represent the system's deployment reliability.

A robustness check excluding the three failures ($n = 47$) confirmed and substantially strengthened the result:

$$\bar{C}_{MAS,clean} = 40.75 = 11.8\ \%. \qquad (10)$$

The Mann-Whitney test on the clean-run MAS sample yielded $\boldsymbol{U = 555.0, p = 7.76 \times 10^{-6}, r = 0.528}$ (large effect), indicating that the MAS advantage is genuine and not an artefact of the distributional asymmetry introduced by failures.

Distance, Energy, and Mission Time Overhead

Total distance travelled was significantly lower in the MAS condition (mean 21.040 ± 7.517 grid units) compared to the no-MAS baseline (24.438 ± 1.472 grid units), with $U = 1.734, p = 8.59 \times 10^{-4}, r = -0.387$ (medium, direction no-MAS > MAS). An equivalent significant difference was observed for energy consumption: 504.65 ± 180.12 Wh (MAS) versus 586.06 ± 35.33 Wh (no-MAS). As established earlier, both differences are predominantly explained by the three zero-distance failure runs that skewed the MAS distribution toward lower values.

The clean-run comparison for these metrics, however, approaches parity. Mission**,** in contrast, time was significantly longer under the MAS condition (mean $\mathbf{64.43 \pm 1.92s}$ versus $\mathbf{60.76 \pm 0.05s}$ for the baseline), with $\boldsymbol{U = 0.0, p = 7.07 \times 10^{-18}, r = 1.00}$ (perfect separation between the two groups). The 3.66s overhead ($\mathbf{6.0}$ % of target mission time) is attributable entirely to the initialisation and operation of the 12 agent threads and their inter-layer message queue.

Search-Space Filtering Behaviour Under Swarm Scaling

The functional analogue of dead-code removal operates over the search space, not the code: the Sensitivity Analyzer filters out hyper parameters whose influence falls below a significance threshold and the Constraint Satisfier discards configurations that violate the physical-stability inequalities, keeping the "live" search space compact.

This filtering is decoupled from the swarm size $\boldsymbol{N}$. Both agents act on the aggregated model from the queen node, whose dimensionality is fixed by the hyperparameter set, so filtering scales as $\boldsymbol{O(P)}$, $\boldsymbol{P = 27}$, rather than $\boldsymbol{O(N)}$. Only the upstream aggregation at the queen is $\boldsymbol{O(N)}$, and it is kept off the critical path by the asynchronous architecture.

As the experiments used a fixed swarm size, this scaling is established analytically rather than measured; an empirical study across varying $\boldsymbol{N}$ is left for future work.

CONCLUSIONS

The experimental results confirm the central hypothesis of this study: the addition of the 12-agent MAS tuning superstructure produces a statistically significant improvement in area coverage relative to the unaugmented chaotic algorithm ($U = 705.0, p = 1.74 \times 10^{-4}, r = 0.436$, medium effect, Table 4). When the three agent initialisation failures are excluded, this effect grows to large magnitude ($r = 0.528, p = 7.76 \times 10^{-6}$), demonstrating that the framework's conceptual architecture – the continuous pipeline – is capable of producing meaningful performance gains when operating correctly.

The results equally demonstrate that the present implementation does not yet constitute a production-ready solution. The primary limitation is the 2.9 × increase in outcome variance (Fig. 5), which stems from two distinct and separable sources:

1. Threading race condition (6 % failure rate): a purely engineering deficiency remedied by adding a synchronisation barrier to MAS-Orchestrator that blocks until the first telemetry packet populates the monitoring queue, guaranteeing drones are active before agents begin processing.

2. Domain-agnostic tuning heuristics: the Exploration Agent and Exploitation Agent employ general-purpose genetic search and hill-climbing, respectively. In a 60-second mission window, the Performance Analyzer accumulates at most 60 one hertz metric samples, which is insufficient for the

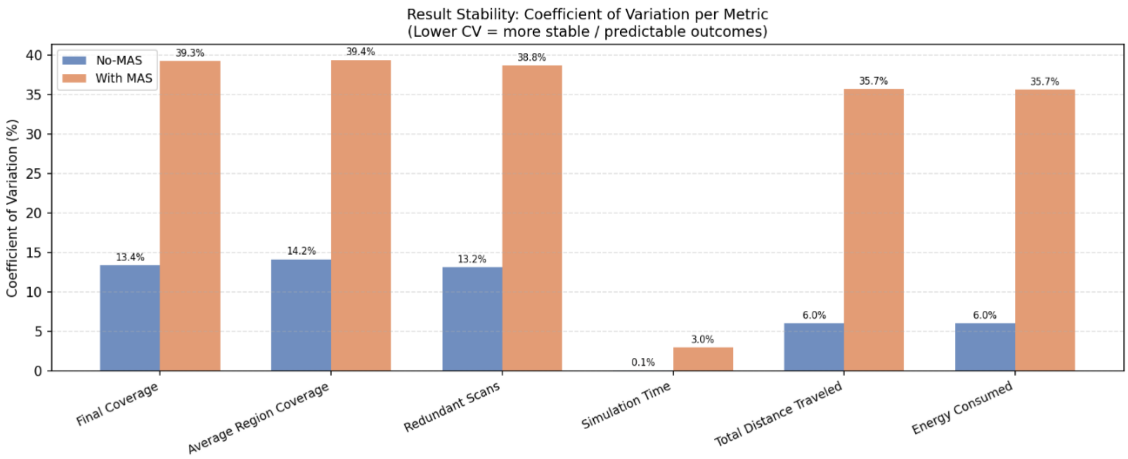


Fig. 5. **Coefficient of Variation per metric for both conditions.**
***Source*: compiled by the authors**

Sensitivity Analyzer to reliably rank parameter influence before the Exploration Agent begins issuing proposals. The result is structured noise injection into an algorithm already sensitive to parameter perturbation, producing high variance even in the absence of failures.

Distanc: without it, MAS-driven proposals led to swarm cluster collapse and coverage below 10% in the majority of runs. The GA effectively constrains the MAS search to a safe neighbourhood Θ_{safe} of Θ^*_{GA}. This relationship suggests a general design principle for systems of this class: a one-time offline optimiser must define the admissible region for online agents, rather than permitting unconstrained exploration of the full hyperparameter space during an active mission.

Achieving statistically reliable and consistent improvement requires addressing two interconnected deficiencies. First, the general-purpose nature of the tuning agents must be replaced with domain-specific knowledge. Training a surrogate model – a Gaussian process regressor or a lightweight neural network – on a corpus of ≥ 500 historical simulation runs would allow the Exploration Agent to generate informed candidate configurations from the first seconds of a mission, rather than performing blind genetic mutation. Such a model learns the non-stationary sensitivity profile of the chaotic algorithm's parameter space, enabling effective proposals well within the convergence time of a 60-second mission. Second, the Exploitation Agent should be initialised from Θ^*_{GA} with its mutation radius restricted to ± 10 % of GA-evolved values during the first 30 seconds of flight, to prevent premature destabilisation of an already near-optimal configuration.

This study establishes both the scientific feasibility and the current engineering limitations of real-time MAS-based hyperparameter tuning for chaotic UAV swarm algorithms. The proposed architecture demonstrates a statistically significant capacity to improve territorial coverage, validated by non-parametric testing across 100 independent simulation runs. The observed instability and overhead are attributable to specific, identifiable deficiencies – a threading initialisation race condition and the absence of a domain-trained proposal model – rather than to any fundamental limitation of the multi-agent iterative tuning paradigm. The integration of a custom-trained surrogate model into the Exploration Agent and the implementation of proper initialisation synchronisation represent the primary directions for future development of this technology.

REFERENCES

1. Chung, S. J., Paranjape, A. A., Dames, P., Shen, S. & Kumar, V. "A survey on aerial swarm robotics". *IEEE Transactions on Robotics*. 2018; 34 (4): 837–855. DOI: https://doi.org/10.1109/TRO.2018.2857475.

2. Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E. & Vicsek, T. "Optimized flocking of autonomous drones in confined spaces". *Science Robotics*. 2018; 3 (20): eaat3536, https://www.scopus.com/pages/publications/85056570738. DOI: https://doi.org/10.1126/scirobotics.aat3536.

3. Yuan, J., Liu, Z., Lian, Y., Chen, L., An, Q., Wang, L. & Ma, B. "Global optimization of UAV Area coverage path planning based on good point set and genetic algorithm". *Aerospace*. 2022; 9 (2): 86, https://www.scopus.com/pages/publications/85124372245. DOI: https://doi.org/10.3390/aerospace9020086.

4. Shao, S., Peng, Y., He, C. & Du, Y. "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization". *ISA Transactions*. 2020; 97: 114–130, https://www.scopus.com/pages/publications/85070395947. DOI: https://doi.org/10.1016/j.isatra.2019.08.018.

5. Sayama, H. "Swarm chemistry". *Artificial Life*. 2009; 15 (1): 105–114. DOI: https://doi.org/10.1162/artl.2009.15.1.15107.

6. Gershenson, C., Trianni, V., Werfel, J., Sayama, H. "Self-Organization and Artificial Life". *Artificial Life*. 2020; 26 (3): 391–408. DOI: https://doi.org/10.1162/artl_a_00324.

7. Zhang, Y., Ding, M., Zhang, J., Yang, Q., Shi, G., Lu, M., & Jiang, F. "Multi-UAV pursuit-evasion gaming based on PSO–M3DDPG schemes". *Complex & Intelligent Systems*. 2024; 10: 6867–6883. DOI: https://doi.org/10.1007/s40747-024-01504-1.

8. Rudenko, O., Bezsonov, O., & Lebediev, O. "Adaptive control over non-linear objects using the robust neural network FCMAC". *Eastern-European Journal of Enterprise Technologies*. 2018; *2*(4) (92), 4–14, https://www.scopus.com/pages/publications/85046084411. DOI: https://doi.org/10.15587/1729-4061.2018.128270.

9. Jayaweera, H. M. P. C. & Hanoun, S. "Path planning of Unmanned Aerial Vehicles (UAVs) in windy environments". *Drones*. 2022; 6 (5): 101. DOI: https://doi.org/10.3390/drones6050101.

10. Kozlov, M. S. & Shvets, Y. O. “Heterogeneous swarm complex multilevel reconfiguration based on Swarm Chemistry”. *Applied Aspects of Information Technology*. 2025; 8: 147–161. DOI: https://doi.org/10.15276/aait.08.2025.10.

11. Hopkins, J., Joy, F., Sheta, A., Turabieh, H. & Kar, D. “Path planning for indoor UAV using A* and late acceptance hill climbing algorithms utilizing probabilistic roadmap”. *International Journal of Engineering & Technology*. 2020; 9(4): 857–862. DOI: https://doi.org/10.14419/ijet.v9i4.31033.

12. Dorri, A., Kanhere, S. S. & Jurdak, R. “Multi-agent systems: A survey”. *IEEE Access*. 2018; 6: 28573–28593. DOI: https://doi.org/10.1109/ACCESS.2018.2831228.

13. Chang, X., Ye, L., Ma, L. & Chen, S. “Research on hybrid path planning algorithms for UAVs in complex environments”. *EAI Endorsed Transactions on Scalable Information Systems*. 2023. DOI: https://doi.org/10.4108/eetsis.8974.

14. Kozlov, M. S. & Malakhov, E. V. “A complex model of human recognition in emergencies by a multi-level drone system”. *International Symposium ELMAR*. 2024. p. 317–320, https://www.scopus.com/pages/publications/85207193007.
DOI: https://doi.org/10.1109/ELMAR62909.2024.10694255.

15. 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/pages/publications/85204938525. DOI: https://doi.org/10.1145/3674558.3674603.

16. Santos, H. V. F. Costa, V., Montandon, J. E. & Valente, M. T. “Decoding the configuration of AI coding agents: Insights from Claude Code Projects”. *Software Engineering (cs.SE)*, 2025. DOI: https://doi.org/10.48550/arXiv.2511.09268.

17. Ferrag, M. A., Lakas, A., Tihanyi, N. & Debbah, M. “LLM and AI Agents for autonomous systems: A survey of applications, datasets, and security challenges”. *IEEE Open Journal of Intelligent Transportation Systems*. 2026; 7: 615–657. DOI: https://doi.org/10.1109/OJITS.2026.3665677.

18. Okoye, K. & Hosseini, S. “Mann-Whitney U Test and Kruskal-Wallis H Test Statistics in R”. *R Programming.* Springer, Singapore. 2024. DOI: https://doi.org/10.1007/978-981-97-3385-9_11.

19. MacFarland, T. W. & Yates, J. M. “Mann-Whitney U Test”. *Introduction to Nonparametric Statistics for the Biological Sciences Using R. Springer*. 2016. p.103–132. DOI: https://doi.org/10.1007/978-3-319-30634-6_4.

20. Kozlov, M. S., Malakhov, E. V. & Shvets, Y. O. “Comparative analysis of chaotic and deterministic methods to territory coverage by drone swarms”. *Herald of Advanced Information Technology*, 2025; 8 (4): 447–463, https://www.scopus.com/pages/publications/105026617701.
DOI: https://doi.org/10.15276/hait.08.2025.28.

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 07.04.2026
Received after revision 10.06.2026
Accepted 17.06.2026

DOI: https://doi.org/10.15276/aait.09.2026.21
УДК 681.5:004.89

Штучний інтелект у задачах оптимізації параметрів багатоагентних систем: модель ітеративного тюнінгу гіперпараметрів в алгоритмах ройового інтелекту

Козлов Микита Сергійович[1)]
ORCID: https://orcid.org/0009–0007–0071–7748; nikita.kozlov@stud.onu.edu.ua
Малахов Євгеній Валерійович[1)]
ORCID: https://orcid.org/0000–0002–9314–6062; eugene.malakhov@onu.edu.ua. Scopus Author ID: 56905389000
Саміканну Раві[2)]
ORCID: https://orcid.org/0000-0002-6945-6562; samikannur@biust.ac.bw

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, вул. Змієнка Всеволода, 2. Одеса, 65082, Україна
[2)] Ботсванський міжнародний університет науки і технологій. Палап'є, Ботсвана

АНОТАЦІЯ

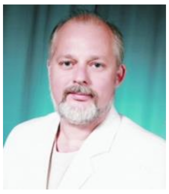
Актуальність дослідження зумовлена необхідністю швидкого та безпечного аналізу територій, що постраждали внаслідок надзвичайних ситуацій, яка стимулює пошук інноваційних підходів у сфері автономних систем, зокрема використання роїв безпілотних літальних апаратів для сканування місцевості. **Метою статті** є представлення інтелектуальної багатоагентної надбудови для динамічного коригування параметрів існуючої ройової системи, що базується на поєднанні механізмів самоорганізації Swarm Chemistry та глобальної оптимізації за допомогою еволюційного алгоритму. **Завдання** полягає в розробці технології ітеративного тюнінгу, яка дозволяє в режимі реального часу адаптувати поведінку окремих агентів та всієї формації у відповідь на динамічні зміни середовища без втручання в базову структуру алгоритму керування. **Серед використаних методів** – архітектура із набору спеціалізованих агентів, організованих у чотири функціональні шари, що здійснюють безперервний моніторинг, аналіз чутливості та коригування гіперпараметрів місії, разом із комбінованою методологією генетичного алгоритму та методів локальної оптимізації, що враховують гетерогенність рою та індивідуальні ширини сканування кожного пристрою. **Наукова новизна** роботи полягає в архітектурі системи динамічного тюнінгу, яка забезпечує живучість та високу продуктивність існуючих ройових рішень у складних умовах. **Практична значимість** визначається значним потенціалом застосування в моніторингу зон лиха, пошуку постраждалих та оцінці руйнувань у реальному часі. **Результати** моделювання продемонстрували значне скорочення часу виконання місії та максимізацію зони покриття при одночасному підвищенні стабільності поведінки рою навіть за умов часткової втрати агентів або появи непередбачуваних перешкод. **Висновки** вказують на подальший розвиток моделі, який передбачає інтеграцію методів машинного навчання для поглибленого аналізу досвіду попередніх місій та розширення системи для роботи у тривимірному просторі.

Ключові слова: інформаційні технології; ройовий інтелект; ітеративний тюнінг; багатоагентні системи; гіперпараметри; хімія рою; генетичний алгоритм; оптимізація сканування

ABOUT THE AUTHORS

Mykyta S. Kozlov - PhD student of Computer Science. Odesa I. I. Mechnikov National University, 2, Vsevolod Zmiienko 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, вул. Змієнка Всеволода, 2. Одеса, 65082, Україна. Senior Software Development Engineer (Back-End), SPD-Tech, Черкаси, Україна



Eugene V. Malakhov - Doctor of Engineering Sciences, Professor, Head of Department of Mathematical Support of Computer Systems. Odesa I. I. Mechnikov National University, 2, Vsevolod Zmiienko Str. Odesa, 65082, Ukraine
ORCID: https://orcid.org/0000-0002-9314-6062; eugene.malakhov@onu.edu.ua; Scopus Author ID: 56905389000
***Research field*:** Databases theory, Metamodeling, Data Mining and other data structuring methods, Data Processing methods, Artificial Intelligence and Machine Learning

Малахов Євгеній Валерійович – доктор технічних наук, професор, завідувач кафедри Математичного забезпечення комп'ютерних систем. Одеський національний університет імені І. І. Мечникова, вул. Змієнка Всеволода, 2. Одеса, 65082, Україна

Ravi Samikannu - Associate Professor, Department of Electrical and Communications Systems Engineering. Botswana International University of Science and Technology.
ORCID: https://orcid.org/ 0000-0002-6945-6562; samikannur@biust.ac.bw
Research field: power systems, renewable energy, intelligent algorithms, and biomedical image processing

Саміканну Раві - доцент кафедри Електротехнічних та комунікаційних систем Міжнародного університету науки і технологій. Ботсвани

Yuliia O. Shvets - Master's student of Information Systems and Technologies. Odesa I. I. Mechnikov National University, 2, Vsevolod Zmiienko Str. Odesa, 65082, Ukraine. Data Quality Engineer at EPAM Systems Inc.
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 - студентка магістратури з Інформаційних систем та технологій. Одеський національний університет імені І. І. Мечникова, вул. Змієнка Всеволода, 2. Одеса, 65082, Україна