

Efficient Exploration in Unknown Environments: An Adaptive Multi-Agent Approach

Daniel Soto¹ and Wilson Soto²

¹ `daniel.soto_forero@univ-fcomte.fr`
FEMTO-ST Institute, DISC Department
Université Marie & Louis Pasteur
Besançon, France

² `wsoto@poligran.edu.co`
ETIC Research Group
Institución Universitaria Politécnico Grancolombiano
Bogotá, Colombia

Abstract. This paper presents an agent-based algorithm for exploring enclosed and uncharted environments, with the primary objective of locating scattered targets. The algorithm addresses key limitations of swarm agents, including restricted perception, limited communication, and constrained memory. It incorporates four exploration strategies – inertia action, virtual attractive force, virtual repulsive force, and random selection – allowing agents to adapt by randomly selecting one at each iteration. The algorithm is evaluated in two simulated scenarios featuring varying agent types and environmental obstacles. Results demonstrate its effectiveness: the swarm consistently achieves target localization while covering approximately 80% of the environment in fewer than 5000 iterations. These findings highlight the algorithm’s capability to enable efficient exploration through decentralized decision-making under partial information. The proposed approach offers a promising alternative for autonomous exploration tasks involving multi-agent systems in constrained or dynamic environments.

Keywords: Agent-based algorithms, autonomous agents, exploration strategies, swarm robotics

1 Introduction

Autonomous navigation and exploration in uncharted environments remain major challenges for intelligent robotic systems. These tasks require the creation of accurate spatial maps to identify enclosed or previously inaccessible regions, alongside the ability to reach designated targets while dynamically avoiding both static and mobile obstacles. Such missions are inherently complex due to diverse scenarios and structural uncertainties [1]. Tackling these challenges frequently necessitates the deployment of a collaborative swarm comprising multiple robots, a strategy proven to substantially reduce completion time in comparison to relying on a solitary robot [2].

While coordinating multiple robots for search and exploration missions, it is essential to consider additional constraints beyond task planning. These constraints include energy limitations, travel costs, and each robot’s restricted sensor capabilities or vision. Moreover, maintaining connectivity among robots and facilitating information transmission is crucial due to the inherent limitations of their sensors [3], [4]. Careful consideration of these factors is imperative for successful multi-robot coordination in such endeavors.

Multi-Agent Systems (MAS) provide a powerful framework for evaluating strategies that involve movement, communication, and synchronization among robots. MAS are well-suited for modeling and simulating swarms, as they can account for the unique characteristics of each robot. They enable the observation of emergent behaviors resulting from information exchange and decentralized decision-making based on each robot’s limited knowledge [5]. Moreover, MAS excel at performing complex tasks in a parallel, collaborative, and fault-tolerant manner [6].

Despite significant progress in swarm coordination and mapping strategies, existing approaches often assume idealized communication or sensor models and may not perform efficiently in highly constrained or dynamic environments. In this paper, we propose a decentralized agent-based algorithm that leverages four adaptable exploration strategies to enable autonomous multi-agent systems to effectively explore and map unknown, enclosed environments while detecting scattered targets. Our method specifically addresses limitations related to partial observability, limited memory, and restricted inter-agent communication.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 presents a detailed description of the proposed multi-agent algorithm. Section 4 covers the experimental setup, including test scenarios, performance metrics, and results. Finally, Section 5 concludes the paper and outlines directions for future research.

2 Related Works

In [8], multi-robot systems are primarily employed for surveillance and exploration tasks. The study focuses on optimizing a drone’s trajectory across various waypoints within an unknown environment. This is achieved by defining a cost function and generating a path using Bézier curves. A key aspect of the work is the incorporation of dynamic obstacles, which necessitates accurate prediction of their future positions to ensure collision avoidance. Experimental results across multiple scenarios demonstrate significant improvements in energy efficiency.

Hartuv *et al.* [2] present a comparative analysis of four algorithms used by a team of robots for collaborative exploration and mapping of an unknown environment. The primary objective is to minimize exploration time while maximizing spatial coverage. A key challenge addressed in the study is decision-making under incomplete information. The authors also emphasize the importance of coordinating individual robot actions to improve overall efficiency and reduce the total exploration time.

In [4], focus on persistent surveillance using multi-robot UAV systems operating under connectivity constraints, where robots must periodically revisit designated sensing locations. The authors propose a hybrid algorithm that integrates three heuristics, modeling the environment as a discretized grid and representing robot movement through graph structures. Their results indicate improved time efficiency, particularly when the number of robots is appropriately matched to the scenario’s size and complexity. However, a key limitation of the approach is its reliance on flawless performance from all robots in the swarm throughout the mission.

Recent research has introduced adaptive algorithms for UAV deployment in dynamic environments. An approach for utilizing UAVs to explore cloud formations is presented, aiming to enhance the understanding of atmospheric phenomena. The algorithm enables real-time trajectory adjustments based on meteorological conditions and predefined exploration patterns. While quantitative performance evaluation is challenging due to the task’s inherent complexity, the collected data significantly contributes to atmospheric modeling and prediction [7]. In [1], the authors investigate autonomous UAV navigation in cave environments, offering critical support for spelunkers and first responders. Originally designed for subterranean exploration, the system adapts to various unknown underground scenarios. It combines mapping, localization, path planning, control, and decision-making, with a collaborative strategy that overcomes sensor limitations, proving robust in the DARPA Subterranean Challenge.

Restrepo *et al.* [9] propose a framework based on nonlinear multi-agent systems to address the rendezvous problem and consensus control in autonomous UAV swarms. The framework computes trajectories for six agents to achieve a predefined formation, which is evaluated through a numerical simulation. Results demonstrate its effectiveness in maintaining the desired formation, avoiding collisions, and preserving continuous inter-agent connectivity. This ability to ensure seamless information exchange highlights the framework’s potential for enabling safe and efficient operations in swarm-based UAV missions.

In [10], the authors introduce an algorithm for autonomous vehicle navigation across varied terrains and complex environments, utilizing multi-agent systems and potential fields. The algorithm accounts for key constraints such as vision and communication ranges to guide agents in obstacle-free scenarios. Each agent independently computes its trajectory and shares it with neighboring agents, fostering a collaborative approach to forming a cohesive group trajectory. Validation in simulated environments with three and ten agents demonstrates the algorithm’s effectiveness in producing accurate trajectories while maintaining the required communication links.

The literature highlights various approaches in multi-agent systems for autonomous navigation, including UAVs and swarm robotics, with a focus on challenges like energy efficiency and obstacle avoidance. However, few studies integrate diverse exploration strategies within a decentralized framework. This gap motivates the proposed algorithm, which aims to enhance swarm adaptability and performance in uncharted environments.

3 Proposed Approach

The proposed multi-agent algorithm involves a swarm of N autonomous robots capable of decentralized exploration in uncharted 2D environments. These robots – such as wheeled or hexapod platforms – are categorized as autonomous mobile units with precise trajectory control, commonly referred to as UGVs (Unmanned Ground Vehicles) or UAVs (Unmanned Aerial Vehicles). Each robot operates under specific constraints, including limited communication range, visual perception, and memory capacity. It is assumed that all robots can traverse the entire spatial domain without being hindered by terrain features or locomotion limitations.

Each robot, denoted as R , operates under a memory constraint r_m , which limits the number of past positions it can store. Additionally, each robot is equipped with a perception field r_p , defining the area it can navigate, and a communication field r_c , enabling information exchange with nearby swarm members. Both r_p and r_c are radial and independent variables, meaning they can be equal or differ in range. The perception field may be larger or smaller than the communication field (see Fig. 1). Communication among agents follows a peer-to-peer model without intermediaries. As a result, the swarm forms a dynamic and random network topology, where communication links are established exclusively between agents within mutual communication range.

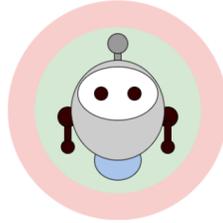


Fig. 1. Visualization of an individual agent within the multi-agent swarm, illustrating its sensing radius (green area) that defines the field of vision, and its communication radius (red area) that determines peer-to-peer connectivity capabilities.

Employing a swarm enables robust, collaborative, fault-tolerant, and parallel exploration of the environment. In this approach, a decentralized architecture is adopted, wherein each agent determines its actions based on local sensor data and information exchanged with neighboring agents. The algorithm is designed to preserve agent autonomy, allowing each unit to operate with individualized parameters, thereby supporting heterogeneous behaviors within the swarm.

Trajectories are computed locally within each agent’s perception radius. To navigate the environment, each agent requires a defined direction and magnitude of movement. Displacement occurs at every iteration of the algorithm, following a straight-line path. Four distinct movement strategies are employed: random

displacement, inertia (maintaining the previously selected direction), and virtual attractive and repulsive forces, which simulate interactions with other agents or environmental features.

In the random displacement strategy, each agent selects a direction θ_c uniformly from the range $[0^\circ, 360^\circ]$, along with a displacement distance d . This distance must be greater than zero and confined within the robot's perception range, ensuring the agent can evaluate whether the intended trajectory is free of obstacles before proceeding. In the case of the inertia strategy, the agent retains its current direction θ_c and varies only the displacement distance d during the subsequent iteration.

Virtual forces are computed based on the aggregate distances to other swarm members, using positional data stored in each robot's local memory. The nature of the displacement – attractive or repulsive – is determined by the selected direction θ_c . For attractive displacement, the direction guides the agent toward neighboring agents, aiming to minimize the inter-agent distance, as defined in Equation (1). Conversely, for repulsive displacement, the agent moves away from others, seeking to maximize this distance, as specified in Equation (2). The function D represents the Euclidean distance between two given points.

$$\theta_c = \arg \min \theta \sum_{i=1}^N D(R_i, r_p \cos \theta), \quad 0 \leq \theta \leq 360 \quad (1)$$

$$\theta_c = \arg \max \theta \sum_{i=1}^N D(R_i, r_p \cos \theta), \quad 0 \leq \theta \leq 360 \quad (2)$$

The selection of displacement strategies is governed by principles of individuality, variability, and controlled randomness. Each robot independently defines lower and upper probability bounds for the four available strategies – repulsive virtual force, attractive virtual force, inertial movement, and random selection – within a normalized global range from 0 to 1. This flexible mechanism allows the swarm to exhibit diverse probabilistic behaviors, fostering heterogeneity in agent actions.

Once the direction θ_c and displacement distance d are determined, the robot's next position (x_{n+1}, y_{n+1}) is calculated by applying the offset from the current position (x_n, y_n) , as defined in Eq.(3).

$$\begin{cases} x_{n+1} = x_n + d \cos \theta_c \\ y_{n+1} = y_n + d \sin \theta_c \end{cases} \quad 0 < r < r_p, \quad 0 \leq \theta_c \leq 360 \quad (3)$$

The primary objective of the robots is to locate all predefined targets within the environment. The total number of targets, denoted by T , corresponds to a set of uniformly distributed waypoints throughout the mapped area. Each robot maintains a local binary vector v_t , indicating which targets it has successfully reached. At the beginning of the exploration process, all entries in v_t are initialized to zero. When a robot encounters a target, it updates the corresponding entry in its vector to one and propagates this information to neighboring agents

within communication range. Consensus among all swarm members – signifying the completion of the exploration task – is achieved once each robot’s v_t vector contains only ones. Algorithm 1 outlines the steps of the proposed method.

Algorithm 1 Proposed algorithm

Input: N, T, r_c, r_p, r_m
Output: $iterations, posVisitedPct$

```

1: for each agent in  $N$  do // Initialization
2:    $v_{t\{0..T\}} \leftarrow 0$ 
3:    $v_{x,y\{0..r_m\}} \leftarrow (0, 0)$ 
4:    $(x, y) \leftarrow (0, 0)$ 
5:    $\theta_c \leftarrow X_{[0..360]}$ 
6:    $iterations \leftarrow 0$ 
7: end for
8: while not consensus do
9:   for each agent in  $N$  do
10:     $action \leftarrow X_{[1..4]}$ 
11:     $d \leftarrow X_{[0..r_p]}$ 
12:    if  $action = 1$  then // Inertia action
13:       $w \leftarrow \theta_c$ 
14:    else if  $action = 2$  then // Virtual attractive force
15:       $w \leftarrow Eq.(1)$  // Compute the swarm’s median direction
16:    else if  $action = 3$  then // Virtual repulsive force
17:       $w \leftarrow Eq.(2)$  // Compute the swarm’s median opposed direction
18:    else if  $action = 4$  then // Random selection
19:       $w \leftarrow X_{[0..360]}$ 
20:    end if
21:     $(x', y') \leftarrow Eq.(3)$ 
22:    if  $(x', y') \notin v_{x,y}$  & avoiding obstacles then
23:       $\theta_c \leftarrow w$ 
24:       $v_{x,y}.add(x', y')$ 
25:       $v_{t\{0..T\}} \leftarrow v'_{t\{0..T\}}$  // Exchange target data with neighboring agents
26:       $iterations \leftarrow iterations + 1$ 
27:    end if
28:  end for
29: end while
30:  $posVisitedPct \leftarrow positionsVisited / (totalPossiblePositions \times 100)$ 
31: return  $iterations, posVisitedPct$ 

```

4 Experimental Results

A series of experimental configurations were systematically tested to evaluate the performance of the proposed algorithm. Simulations were conducted with fixed parameters: communication radius $r_c = 5$, perception radius $r_p = 5$, marking radius $r_m = 10$, and a time threshold $T = 10$, while the number of agents

N varied across different configurations. Each setup was evaluated over 20 independent simulation runs in two distinct experimental scenarios, as depicted in Figure 2).

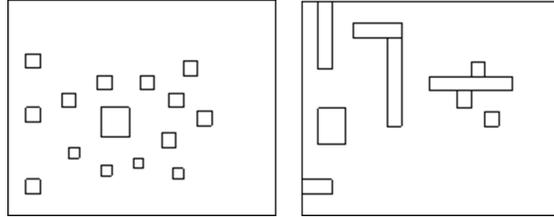


Fig. 2. Simulated environments for Scenario 1 and Scenario 2. Although the layouts differ, both scenarios occupy an area of $22325 \text{ mm} \times 18800 \text{ mm}$. Robots in the simulation have a diameter of approximately 470 mm .

A comprehensive overview of the numerical results obtained across simulated scenarios is presented in Table 1. Agents in all simulations started from the same position. Figure 3 shows exploration over iterations in both scenarios.

Table 1. Performance metrics for Scenarios 1 and 2: mean number of iterations to locate all targets (MIT), achieve consensus (MIC), and environment coverage (MPE).

Scenario 1						
N	MIT	MIC	MPE	MIT/ N	MIC/ N	MPE/ N
5	2460	649.95	94.894	492	129.99	18.978
10	1110	381.80	94.361	111	38.18	9.4361
20	805	247.20	96.606	40.25	12.36	4.8303
50	790	287.50	97.712	15.80	5.75	1.9542
500	432	30.20	96.784	0.864	0.0604	0.1935
1000	404	40.15	96.265	0.404	0.04015	0.0962
Scenario 2						
N	MIT	MIC	MPE	MIT/ N	MIC/ N	MPE/ N
5	2545	656.80	92.144	509	131.36	18.428
10	1205	598.80	93.384	120.5	59.88	9.338
20	845	385.85	95.664	42.25	17.942	4.783
50	510	169.25	95.406	10.2	3.385	1.908
500	438	31.20	95.116	0.876	0.0624	0.190
1000	446	31.10	95.608	0.446	0.0311	0.095

Interestingly, the highest average exploration percentage is not necessarily attained by employing a larger number of agents, as quicker target achievement can offset the benefits of increased swarm size. In scenario 1, the peak exploration percentage is observed when utilizing 50 agents, whereas in scenario 2, the optimal exploration is achieved with a swarm comprising 20 agents.

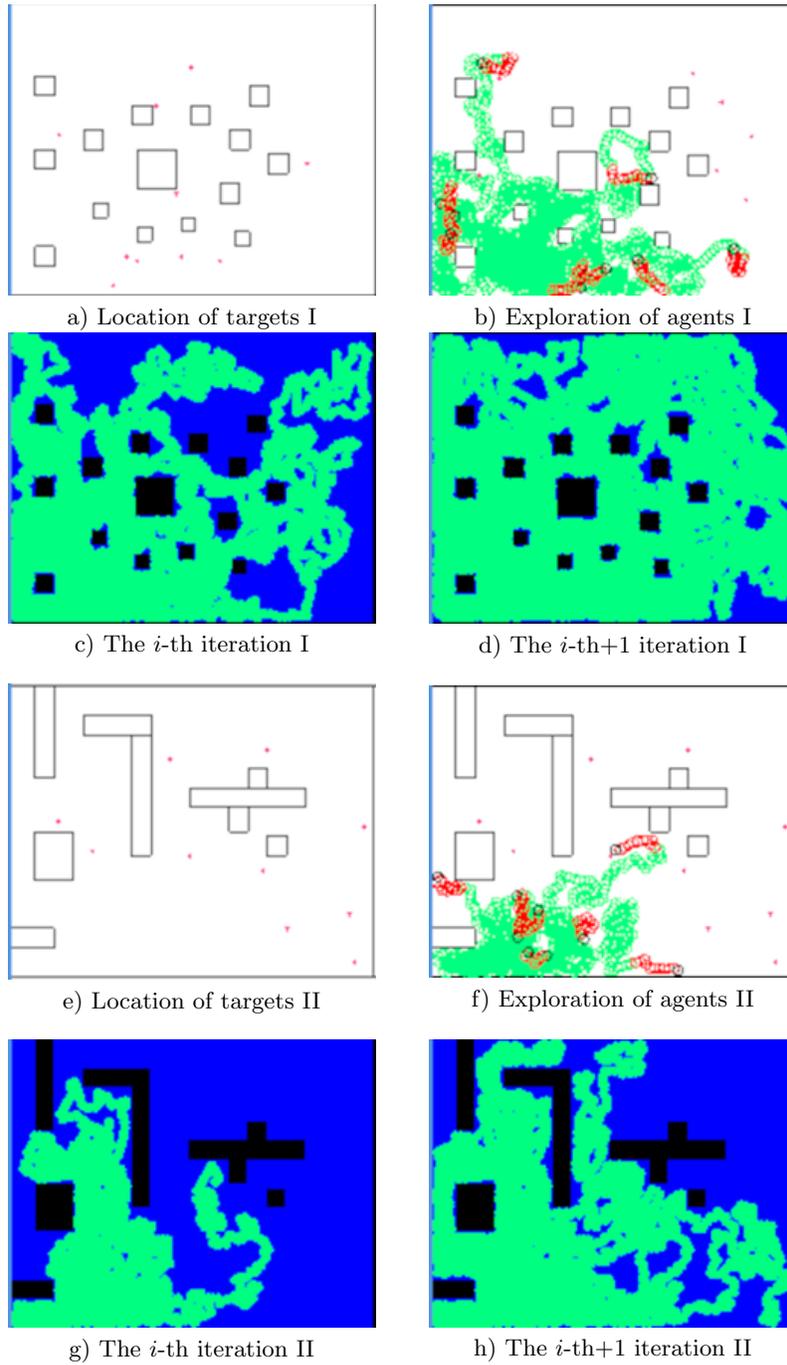


Fig. 3. Exploration behavior of agents across different scenarios. Scenarios 1 (I) and 2 (II) are illustrated in subfigures a–d and e–h, respectively. The subfigures show the initial target locations, agent trajectories (red), highlighting explored areas (green), and unexplored areas (blue).

Figure 4 shows the exploration percentage upon reaching each i -th target for the six configurations in Scenario 1. Similarly, Figure 5 presents the results for Scenario 2. In both cases, the results indicate that identifying a greater number of targets requires agents to explore a larger portion of the environment. Nonetheless, based on median values, configurations with larger agent populations (500 and 1000 agents) are able to locate up to three targets while exploring less than 50% of the scenario.

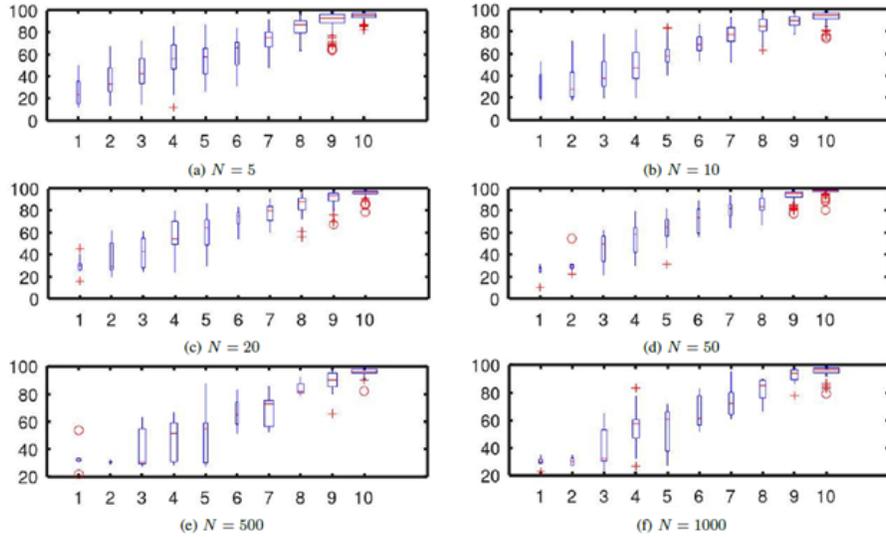


Fig. 4. Correlation between the explored percentage of Scenario 1 and the number of targets.

Figures 6a and 6b show the number of iterations required by the proposed algorithm to complete all objectives in Scenarios 1 and 2, respectively, across six configurations. In both scenarios, the fewest iterations and the best balance between performance and resource usage occur with 500 agents. Although a similar iteration count is achieved with 1000 agents, the computational cost increases significantly with the larger swarm size.

Figure 6c and 6d show the number of iterations required for the proposed algorithm to achieve consensus among all agents in Scenarios 1 and 2, respectively, across the six configurations. The consensus behavior follows a similar trend to the number of iterations. Despite having more agents, which results in longer consensus times due to additional transmission stages, the overall iteration count remains unchanged. This is because a larger agent population covers a broader area, increasing the likelihood of agents being within transmission range. In contrast, with fewer agents, the challenge is finding others within range, requiring more iterations.

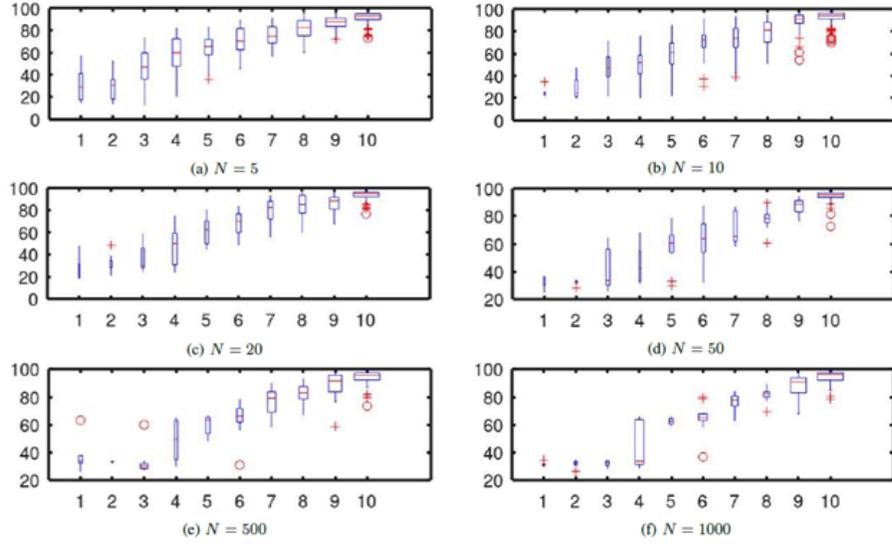


Fig. 5. Correlation between the explored percentage of Scenario 2 and the number of targets.

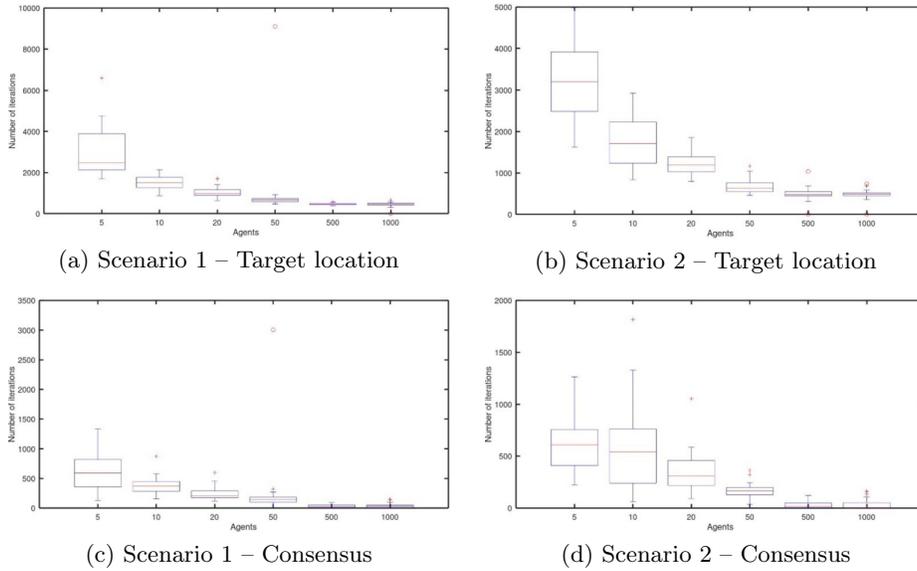


Fig. 6. Iterations required across 20 simulation runs for two performance metrics. (a–b) Number of iterations to locate all targets in Scenarios 1 and 2. (c–d) Number of iterations to reach consensus in Scenarios 1 and 2.

5 Conclusions

The experimental evaluation across simulated scenarios revealed a clear trend: increasing the swarm size led to a significant reduction in the number of iterations required to achieve the predefined objectives. This behavior reflects an asymptotic global pattern, where the swarm's collective actions progressively converge toward consensus.

The proposed algorithm demonstrated robustness to communication failures and agent interruptions, enabling the swarm to reach consensus, achieve its objectives, and complete the exploration of the environment.

Future research could explore test configurations with heterogeneous agents to enable comparative analyses of exploration performance. Additionally, extending the proposed algorithm to operate in three-dimensional environments or those with dynamic, mobile obstacles represents a promising direction for further study.

References

1. Petráček, P., Krátký, V., Petrlík, M., Báča, T., Kratochvíl, R., Saska, M.: Large-Scale Exploration of Cave Environments by Unmanned Aerial Vehicles. In: Proceedings of the IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 7596–7603 (2021). DOI: 10.1109/LRA.2021.3098304
2. Hartuv, E., Agmon, N., Kraus, S.: Spare drone optimization for persistent task performance with multiple homes. In: Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS 2020), pp. 389–397. IEEE, Athens, Greece (2020). DOI: 10.1109/ICUAS48674.2020.9213932
3. Scherer, J., Rinner, B.: Multi-robot persistent surveillance with connectivity constraints. IEEE Access, 8, 134019–134030 (2020). DOI:10.1109/ACCESS.2020.2967650
4. Mohamed, K., Elshenawy, A., Harb, H. M.: An Evaluation of Multi-Robot Systems Exploration Algorithms. Journal of Al-Azhar University Engineering Sector, 14(51), 2019. DOI: 10.21608/aej.2019.33359
5. Kwa, H.L., Leong Kit, J., Bouffanais, R.: Balancing collective exploration and exploitation in multi-agent and multi-robot systems: A review. Frontiers in Robotics and AI, 8, 771520 (2022). DOI: 10.3389/frobt.2021.771520
6. Wang, H., Wong, K.-F.: A Collaborative Multi-agent Reinforcement Learning Framework for Dialog Action Decomposition. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 7882–7889. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (2021). DOI: 10.18653/v1/2021.emnlp-main.62
7. Verdu, T., Bronz, M., Hattenberger, G., Narvor, P., Seguin, F., Lacroix, S., Maury, N., Roberts, G., Couvreur, F., Cayez, G.: Experimental Flights of Adaptive Patterns for Cloud Exploration with UAVs. In: Proceedings of the 2020 International Symposium on Experimental Robotics (ISER 2020), Springer, Cham (2021).
8. Margraff, J.: Planification de mouvement 3D en temps réel de drones autonomes dans des environnements dynamiques inconnus. Ph.D. thesis, Université de Limoges, France (2022).

9. Restrepo, E., Loria, A., Sarras, I., Marzat, J.: Robust consensus of high-order systems under output constraints: Application to rendezvous of underactuated UAVs. *IEEE Trans. Autom. Control* 68(1), 329–342 (2023).
10. Tran, N., Prodan, I., Grøtli, E., Lefèvre, L.: Distributed nonlinear optimization-based control for multi-agent systems navigation in a coastal environment. In: *Proceedings of the 26th European Control Conference (ECC 2019)*, pp. 588–695. IEEE, Naples, Italy (2019).