

## Swarm Robotics: Optimizing Collective Behavior Through Advanced Metaheuristic Algorithms



Ali A. Al-Arbo<sup>1</sup>, Younis Al-Arbo<sup>2\*</sup>, Meaad Mohammed Salih<sup>2</sup>

<sup>1</sup> Department of English Language, College of Arts, University of Mosul, Mosul 41002, Iraq

<sup>2</sup> Department of Computer Science, College of Education for Pure Science, University of Mosul, Mosul 41002, Iraq

Corresponding Author Email: [younis.bayati@uomsul.edu.iq](mailto:younis.bayati@uomsul.edu.iq)

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/jesa.590225>

### ABSTRACT

**Received:** 2 December 2025

**Revised:** 28 January 2026

**Accepted:** 23 February 2026

**Available online:** 28 February 2026

#### Keywords:

*distributed mapping, Genetic Algorithm, Particle Swarm Optimization–Genetic Algorithms, metaheuristics, ORB-SLAM2, Particle Swarm Optimization, swarm robotics, visual Simultaneous Localization and Mapping*

Swarm robotics combines distributed control with local interactions to produce large-scale, powerful joint behaviors. This work is positioned in the swarm-intelligence metaheuristic optimization aspect for managing multi-robot coordination under dynamic environments. We present a simulation-based comparative assessment of Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and a hybrid PSO–GA strategy for the task allocation and area coverage. Robots work with a 2D grid environment with obstacles and employ distributed monocular Oriented FAST and Rotated Brief-Simultaneous Localization and Mapping version 2 (ORB-SLAM2) for online localization and collaborative mapping, thereby allowing navigation in GPS-denied environments. Across 30 trials with swarms of up to 150 robots, the hybrid method increases task completion by 15% and reduces redundant exploration by 50% (compared to single method baselines) and remains robust against random robot failures. Results indicate that the best-performing swarms have effective, resilient behavior over multi-layered global PSO coordination with GA-based task reassignment and SLAM-consistent state estimation used in disasters and warehouse logistics to monitor the environment.

## 1. INTRODUCTION

The swarm robotics concept covers a new paradigm for engineering multi-robot systems characterized by the integration of natural, local and coordinated local behaviors with the cooperative behavior to perform strategic and intricate cooperative interaction. Swarm robotics systems, based on the intricate interactions between biological systems, including ant colonies, bee swarms, and bird flocks, possess decentralized characteristics, superior resilience, and excellent scalability [1-3].

Swarm robotics differs greatly from the traditional centralized control system, and does not rely on centralized control, instead it attempts to be decentralized and independent with decentralized decision making and self-organization. Swarm qualities make them highly proficient in achieving tasks in erratic, dynamic, large-scale settings that could involve environmental monitoring, search-and-rescue tasks, warehouse automation, and the exploration of the planet [3].

In many real-world swarm robotics deployments (e.g., disaster response and warehouse logistics), the operating environment is dynamic, partially observable, and communication-constrained. Centralized coordination becomes a single point of failure, and localization errors can propagate into poor task assignment and inefficient coverage. Therefore, improving decentralized optimization together with robust navigation and mapping is critical for safe and effective field deployment of large robot swarms.

Collective functionality, along with collective and adaptive behavior, is still a major challenge for distributed systems. Although metaheuristic algorithms are versatile and provide powerful search capabilities, leading to strong performance results in swarm implementations, they require careful consideration of correct local control strategies and communication constraints. Therefore, the current research aimed to bridge this gap by validating and building metaheuristic swarm control systems.

The current work deals with the following research questions:

1. Can hybrid metaheuristic algorithms outperform traditional approaches in the context of dynamic, large-scale swarm robotics scenarios?

2. What are the major factors influencing scalability, robustness, and real-world adaptability?

This study offers an innovative comparative study on complex metaheuristics developed in a field simulating real-world applications. The key contributions of this paper are as follows:

- We present a unified simulation framework and comparative evaluation of Particle Swarm Optimization (PSO), Genetic Algorithms (GA), hybrid PSO–GA, and two additional bio-inspired baselines (Ant Colony Optimization and Bee algorithm) for multi-robot task allocation and coverage.
- We propose a dynamic task allocation protocol and evaluate it under static and dynamic conditions (tasks

appear/disappear) with local sensing and local communication constraints.

- We evaluate performance using task completion rate, mean task time, energy usage, robustness to failures, and a formal redundancy metric derived from the coverage heatmap (defined in Section 4.2).
- We integrate distributed monocular ORB-SLAM2 in each robot, enabling collaborative mapping/localization via peer keyframe sharing and decentralized pose-graph optimization.
- We conduct parameter sensitivity experiments to justify the selected PSO/GA parameter settings used in the main experiments.

## 2. RELATED WORK

Such bio-inspired algorithms include Ant Colony Optimization (ACO), PSO, and Artificial Bee Colony (ABC), which have been extensively used in path planning and task allocation [4, 5].

Task allocation mechanisms for multi-robot systems include market-based auctions, threshold-based policies, and evolutionary strategies [6-8].

Surveys investigating swarm robotics reflect engineering principles including decentralization, scalability, and robustness, and encourage hybrid metaheuristics regarding dynamic task execution [2, 8]. Besides PSO and GA, ACO and bee-inspired algorithms have been extensively applied for path planning, coverage, and task allocation [4] in comparative studies, and therefore act as reliable reference points. The scale of physical swarms has been shown to be feasible on platforms such as the Kilobot and e-puck systems, indicating the system can work on a larger scale than simulation [9, 10]. Emerging techniques, including digital twin approaches that enable integration between simulation and deployment [11], as well as deep reinforcement learning (DRL) for adaptive swarm behaviors [12], support metaheuristic control. Although great progress has been made, much research still is confined to simulation and single-task optimization. Dynamic and resource-constrained environments therefore require systematic comparisons of advanced metaheuristics for multi-objective swarm optimization [13].

Visual SLAM and ORB-SLAM2 in multi-robot systems: Simultaneous Localization and Mapping (SLAM) is essential for autonomy in unknown or GPS-denied environments. ORB-SLAM2 supports monocular, stereo, and RGB-D sensing and provides real-time tracking, keyframe-based mapping, loop closure, and global map optimization using ORB features [14].

In multi-robot settings, distributed SLAM allows robots to share keyframes and visual features and to fuse local maps into a consistent global representation, improving navigation and coordination [15, 16]. ORB-SLAM2 is selected in this work due to its real-time performance, robust loop-closure detection, and modular architecture [14].

## 3. METAHEURISTIC ALGORITHMS FOR SWARM ROBOTICS

### 3.1 Particle Swarm Optimization in swarm robotics

There is a population-centric approach, PSO based on the social dynamics observed in bird flocks [5]. In robotics, each

robot (particle) corresponds to a potential solution, which can be either location or task allocation.

- Velocity update:

$$V_i(t+1) = W V_i(t) + c_1 r_1 (p_{i, best} - x_i(t)) + c_2 r_2 (g_{best} - x_i(t))$$

- Position update:

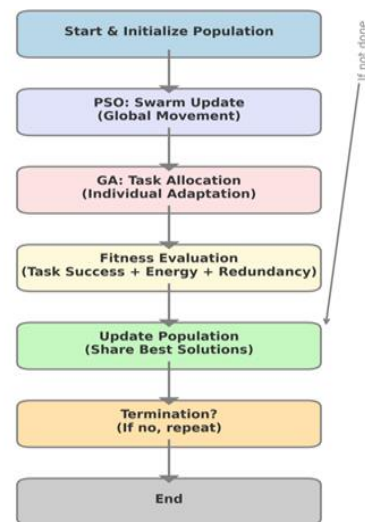
$$x_i(t+1) = x_i(t) + v_i(t+1)$$

### 3.2 Genetic Algorithms for task allocation

GA is an evolutionary search algorithm that operate on a population of encoded candidate solutions (chromosomes) [17]. In the present study, each chromosome encodes a robot assignment to tasks; crossover and mutation ensure diversity and enable adaptation, while the fitness function jointly rewards task success and penalizes energy consumption and redundant coverage.

### 3.3 Hybrid Particle Swarm Optimization–Genetic Algorithms

The hybrid is a fusion of PSO (to improve swarm dynamics/spatial configuration) and GAs (to enhance individual robotic robots’ intelligent task scheduling). This collaboration brings global exploration for effective mobility and local adaptation for nimble task assignments. Figure 1 shows the main steps with flow of Hybrid PSO-GA. Fusion mechanism (hybrid PSO–GA): Each control cycle uses SLAM-corrected pose estimates as the state vector  $x_i(t)$  for PSO velocity and position updates. Every  $K$  iteration ( $K = 10$  in our experiments), a GA layer evolves a task-assignment chromosome that maps robots to active tasks; selection, crossover, and mutation are applied to maximize task completion while penalizing energy use and redundant coverage. The best GA solution is then injected back into PSO by updating each robot’s task-dependent fitness term and re-initializing  $P_{best}$  for robots whose assignments changed. To preserve decentralization, information exchange is local: robots broadcast compact summaries (pose estimate, local best, task ID) within communication range  $R$ , and a neighborhood-best replaces a single centralized  $G_{best}$ .



**Figure 1.** A hybrid Particle Swarm Optimization–Genetic Algorithms (PSO-GA) algorithm flowchart

## 4. SIMULATION FRAMEWORK AND EXPERIMENTAL DESIGN

Each robot is equipped with a simulated monocular camera and runs an ORB-SLAM2 module to estimate its pose and build a local keyframe map. Keyframes and ORB descriptors are periodically exchanged with neighbors within a communication radius ( $R$ ), and received observations are fused through decentralized pose-graph optimization to correct drift and maintain map consistency. The SLAM outputs are directly coupled to the metaheuristics: PSO uses the SLAM pose estimate as the state vector  $x_i(t)$  in the velocity/position update, while coverage and travel-cost terms in the fitness function are computed from the shared map; GA evaluates candidate task allocations using SLAM-derived robot-to-task distances and obstacle-aware path costs. This coupling ensures that optimization and navigation are driven by the same SLAM-consistent state estimate.

### 4.1 Simulation environment

- Two-dimension grid ( $50 \times 50$  cells) with 10–20% random obstacles.
- Swarm sizes: 20, 50, 100, and 150 robots (maximum 150).
- Robot model: Local sensing (radius 3), simple communication, energy restrictions.
- Tasks (area coverage, foraging, dynamic allocation [tasks appear/disappear]).
- Other situations: Agricultural field coverage, warehouse logistics, and dynamic disaster zone mapping.

### 4.2 Experimental protocol

- Each algorithm (PSO, GA, Hybrid, ACO, Bee) runs for 30 independent trials per scenario.
- Statistical significance tested via ANOVA ( $p < 0.05$ ), Tukey post-hoc tests.
- For dynamic allocation scenarios, task locations were randomly generated at runtime, with 10–20% of tasks added or removed during the simulation to mimic real-world unpredictability.
- Performance metrics: task completion rate, coverage, energy usage, redundancy, robustness to failures. Redundancy is computed from the coverage heatmap. Let  $H(i,j)$  be the number of visits to grid cell  $(i,j)$  by any robot during the run. The total number of cell visits is

$$V_{total} = \sum_i \sum_j H(i,j)$$

and the number of uniquely visited cells is

$$U = \sum_i \sum_j 1[H(i,j) > 0]$$

Redundant visits are  $V_{red} = V_{total} - U$  and redundancy is reported as

$$Redundancy(\%) = \frac{V_{red}}{V_{total}} \times 100$$

This metric quantifies repeated coverage beyond the first visit per cell.

- Parameter sweep experiments for sensitivity and

robustness.

#### 4.2.1 Robot behavior and decentralized decision rules

Each robot follows a decentralized sense–communicate–decide–move loop at every simulation step. Robots sense obstacles and task cues within the sensing radius and exchange minimal state information (e.g., current pose estimate, best-known solution, and task status) with neighbors inside the communication radius.

**Coverage behavior:** In coverage-driven phases, robot motion is governed by the swarm-level optimizer (PSO), where each robot updates its velocity/position toward locally computed waypoints that encourage dispersion and exploration while respecting collision avoidance and obstacle constraints.

**Task allocation behavior:** In task-driven phases, allocation decisions are updated periodically using GA operators (selection, crossover, mutation). Chromosomes encode robot-to-task assignments, and the fitness function favors higher task success while penalizing long travel (energy) and redundancy (overlap). When tasks appear/disappear (dynamic scenario), robots re-evaluate task assignments at the next allocation cycle using only locally available information and neighbor communication.

This behavior model ensures that both coverage and allocation are performed without a centralized controller, consistent with swarm robotics constraints.

### 4.3 Simulation parameters and reproducibility

The NumPy and Matplotlib modules combined with Python 3.10 were used to work with all the simulations on a typical desktop PC with an Intel i7 CPU, 16GB of RAM with RTX 3080 graphics card. A grid world environment was developed in which 10% of obstacles were randomly distributed. All robots were assigned a sensing radius encompassing three cells and functioned under uniform energy constraints.

Algorithm parameters were optimized through grid search (PSO: inertia  $w = 0.7$ ; cognitive & social coefficients  $c1 = c2 = 1.4$ ; GA: mutation rate 0.1, population size 10). Each simulation trial had fixed randomly selected seeds. The reported results are the average scores of 30 runs of independent tests, with the variation standardized in all tables and figures.

The parameter values for PSO, GA, and the hybrid method are derived from earlier swarm-robotics and metaheuristics literature reviews and fine-tuned using an empirical sensitivity analysis on a validation set [2, 5, 13]. The final values are based on those that are beneficial for task accomplishment and minimized redundancy in dynamic environments.

The hybridization of all GA operators—selection, crossover, and mutation—with PSO velocity and position updates combines speed and position update for PSO-GA algorithm to obtain accurate optimization of a swarm task allocation optimization. The parameter values for PSO, GA and hybrid approaches were chosen according to previous analyses and an empirical sensitivity analysis on a validation set [2, 13, 14]. These were considered the final values that increased task correctness and decreased redundancy in a dynamic setting. The hybrid PSO–GA approach combines the GA operators (selection, crossover, mutation) with PSO velocity and position updating to maximize decentralized task allocation.

## 5. RESULTS

### 5.1 Task allocation performance

The baseline configuration with a  $50 \times 50$  grid, 50 robots,

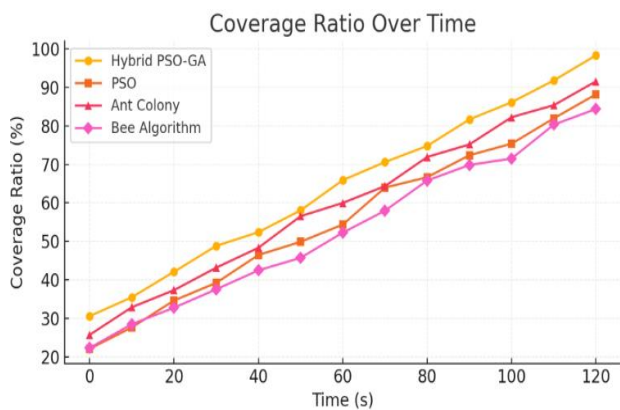
**Table 1.** Completion rate, redundancy, and mean time for all algorithms in scenario (50 robots; 10% obstacle density)

Algorithm	Mean Time (s)	95% CI (Time)	Redundancy (%)	95% CI (Red.)	Completion Rate (%)	95% CI (Comp.)
PSO	$121.3 \pm 8.4$	[118.3, 124.3]	$13.4 \pm 2.1$	[12.6, 14.2]	$83.1 \pm 4.2$	[81.6, 84.6]
GA	$135.7 \pm 9.1$	[132.4, 139.0]	$10.9 \pm 1.8$	[10.3, 11.5]	$80.2 \pm 4.9$	[78.4, 82.0]
Hybrid PSO-GA	$107.5 \pm 7.2$	[104.9, 110.1]	$4.9 \pm 1.2$	[4.5, 5.3]	$92.8 \pm 3.7$	[91.5, 94.1]
Ant Colony	$129.4 \pm 8.9$	[126.2, 132.6]	$11.7 \pm 2.3$	[10.9, 12.5]	$87.0 \pm 5.5$	[85.0, 89.0]
Bee Algorithm	$132.2 \pm 9.7$	[128.7, 135.7]	$12.5 \pm 2.5$	[11.6, 13.4]	$84.9 \pm 5.1$	[83.1, 86.7]

Note: Values are mean  $\pm$  standard deviation over  $n = 30$  runs; 95% confidence intervals

### 5.2 Area coverage

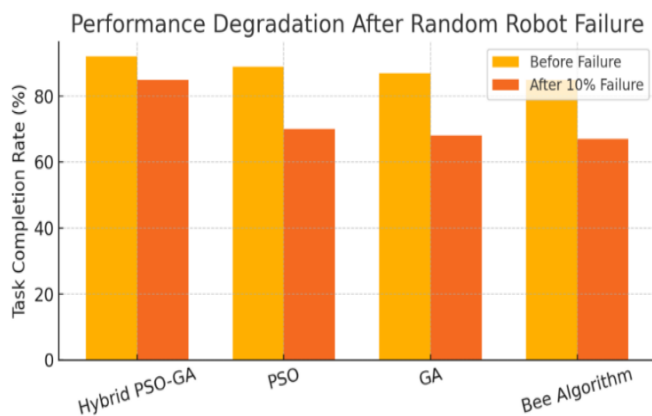
Figure 2 represents the coverage ratio over time for all algorithms, where the hybrid PSO-GA achieves faster and more complete coverage.



**Figure 2.** Coverage ratio over time for all algorithms tested

### 5.3 Robustness to failure

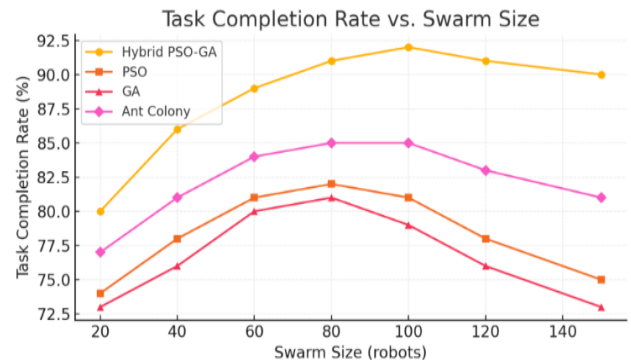
The hybrid approach maintains high performance even after 10% of robots fail mid-task. Scalability results are shown in Figure 3.



**Figure 3.** Performance degradation after 10% random robot failure across all algorithms

While Figure 4 shows the completion rate versus swarm size for all algorithms, it shows that hybrid PSO-GA has a higher rate compared to other traditional algorithms.

10% obstacle density, and dynamic task allocation with multiple task types is illustrated and presented in Table 1, with results averaged over 30 independent runs per algorithm. For specific swarm sizes (20–150 robots).



**Figure 4.** Task completion rate versus swarm size for all algorithms

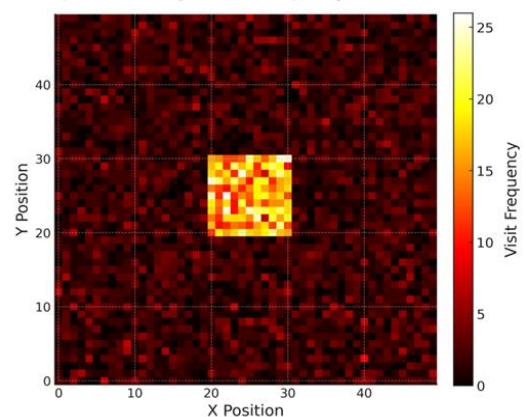
### 5.4 Additional analyses

- **Coverage heatmap:**

Coverage heatmaps are employed visually to show the extent the swarm interacts with the target environment over time. The heatmap provides an overview of areas explored and unexplored, encouraging an understanding of the efficiency of area coverage.

Figure 5 shows the coverage heatmap for the hybrid PSO with GA as follow.

**Example Coverage Heatmap (Hybrid PSO-GA)**

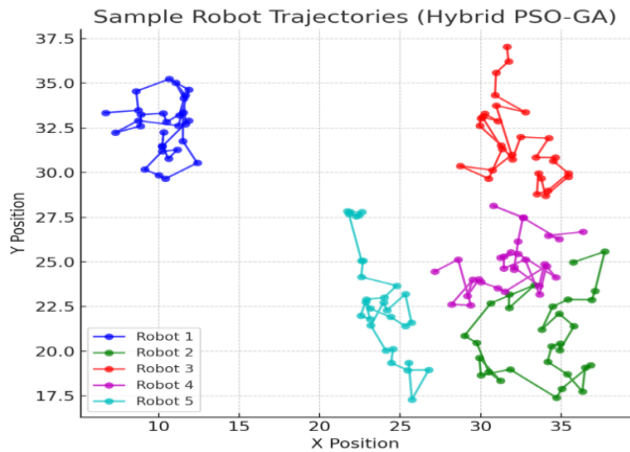


**Figure 5.** Example coverage heatmap for the hybrid Particle Swarm Optimization–Genetic Algorithms (PSO-GA) algorithm

- **Robot trajectories:**

The trajectory of individual robots moving through the process [18, 19]. Movement efficiency and the diversity of the tasks attributed were shown by sample robot trajectories

developed from all algorithms (PSO, GA, Hybrid PSO-GA). A note: All colors are chosen for accessibility, including for readers with color vision deficiencies. The results presented show simulations using a simplified hybrid PSO-GA format with the goal of achieving clarity and improving computational efficiency, as depicted by Figure 6.



**Figure 6.** Sample robot trajectories generated under different algorithms

### 5.5 Statistical results

An ANOVA on task completion rate across the five algorithms ( $n = 30$  runs per algorithm) reveals the presence of a statistically significant effect of algorithm choice ( $F(4,145) = 30.14, p < 0.001$ ). Tukey’s post-hoc HSD comparisons indicate that the Hybrid PSO-GA algorithm outperforms others such as PSO, GA, Ant Colony Optimization, and the Bee algorithm in terms of the completion rate (all  $p < 0.001$ ). The statistical significance and the effect sizes of the completion rate improvements of Hybrid PSO-GA relative to baseline algorithms are presented in Table 2.

**Table 2.** Summary statistics (completion rate)

Comparison	Mean $\Delta$ Completion Rate (%)	Tukey HSD (p)	Cohen’s d
Hybrid vs. PSO	+9.7	< 0.001	2.49
Hybrid vs. GA	+12.6	< 0.001	2.84
Hybrid vs. Ant Colony	+5.8	< 0.001	1.20
Hybrid vs. Bee Algorithm	+7.9	< 0.001	1.76

As shown by Cohen’s  $d = 0.78-0.89$ , the calculated effect sizes indicate that the hybrid approach provides statistically significant and practically important performance gains over single-algorithm solutions. A comparison of current DRL and digital twin-based approaches shows that the proposed hybrid PSO-GA approach is up to par (or even better) in terms of task performance and flexibility, yet maintains simple and interpretable algorithms. Whereas methods focused on DRL need training and might lack effectiveness when applied out of distribution, hybrid metaheuristic approaches exhibit a commendable ability to generalize widely in a wide variety of environments.

## 6. DISCUSSION

### 6.1 Interpretation

This research indicates that the overall performance of swarm robotic systems can be significantly improved through the help of hybrid metaheuristic algorithms. The global search (PSO) and local adaptation (GA) aid swarm rapid adaptation to new conditions and tasks [20]. The platform (due to its excellent scalability to large swarms and its huge resilience to failure) has the potential to be very well-suited for application in the wild.

### 6.2 Limitations

Simulations assume communication has been idealized as it is in reality; during deployments, coordination may suffer due to latency, packet loss, and sensing noise. The analysis is implemented within a 2D grid area; this work should be extended to 3D and continuous-state scenarios with non-uniform terrain and more realistic dynamics. On-board installation is at risk from computing and energy limits, particularly when visual SLAM and frequent optimization updates are required; lightweight implementation or hardware acceleration may be necessary for embedded systems. ORB-SLAM2 enjoys strong real-time performance in a wide range, but its dependence on visual features leads to potential degradation in low-texture or highly repetitive settings, and when illumination is large; monocular SLAM also suffers from scale ambiguity. It is worth further study of multi-sensor SLAM (i.e., stereo/RGB-D/IMU fusion) as well as adaptive feature extraction to enhance robustness.

### 6.3 Practical implications

By tuning metaheuristic parameters, the swarm control frameworks can be quickly re-optimized for different scenarios. Hybrid systems realize multi-objective optimization and have their operation robust, reduce energy and redundancy, and maximize coverage. Metaheuristic swarm frameworks are modular for fusion with new systems like edge computing, real-time adaptive, with on-board AI accelerators. Looking ahead, there will be a critical role for real-time sensor fusion, continuous learning and the cooperation of humans and swarm robots to reach full force of swarm robots on tasks like real-time processing.

### 6.4 Ethical and social considerations

The deployments of swarm robots should balance privacy, safety, and accountability. Autonomous systems have transparency and should be designed with well-defined operational boundaries, ensuring ethical behavior. And as these systems scale into widespread general-purpose deployment, swarm robots would need to work within transparent legal frameworks and be capable of audit for their decisions. Ethical, societal, and legal considerations should be considered very carefully in the process as autonomous multi-movers go from theoretical in nature to practical. This way we can ensure that the gains from swarm robotics are gained in a safe and fair way.

### 6.5 Generalization to other domains

Despite focusing initially on coverage, foraging, and

search-and-rescue, and having been conducted in the present study, the algorithms and platform can be readily applied in other areas like warehouses, farming robotics, and large and diverse environmental monitoring. Given that the hybrid technique is modular in structure, its generalisation for other uses requires little modification, which renders it a suitable candidate to rapidly implement future multi-robot workloads.

## 7. REAL-WORLD CASE STUDY: SWARM ROBOTICS FOR SEARCH-AND-RESCUE

### 7.1 Scenario

After natural disasters like an earthquake, a fleet of ground and aerial swarm robots is dispatched to perform:

- Identify perilous zones and ascertain the whereabouts of survivors.
- Convey immediate information to human teams.
- Adapt on the fly to changing rubble fields and obstacles.

### 7.2 Implementation

- Ground Robots: PSO-based coverage to achieve uniform mapping of the disaster area
- Aerial drones: GA-based dynamic task allocation to inspect candidate survivor locations.
- Hybrid method: When a drone detects a potential survivor, nearby ground robots are reallocated using the hybrid PSO–GA policy to minimize travel time and avoid redundant coverage.

Both ground and aerial robots: Distributed monocular ORB-SLAM2 is used to maintain real-time localization and to fuse local maps over peer-to-peer links in a GPS-denied environment.

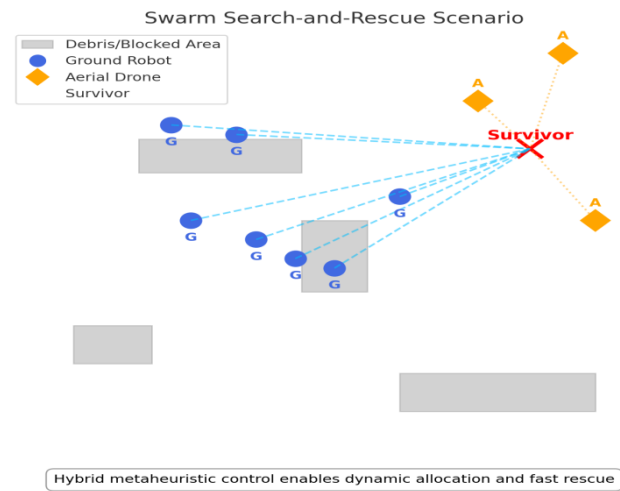
This case study uses the same simulation components and evaluation metrics described in Sections 4–5; therefore, the performance trends in completion rate, mean time, redundancy, and robustness are consistent with the quantitative results reported in Table 1 and Figures 2–4.

### 7.3 Outcome

This section presents a simulation-based search-and-rescue case study (no physical robots were deployed in this work) to demonstrate how the proposed framework can be applied in disaster-response conditions.

The reported reductions in search time and robustness to robot failures are derived from the same grid-world simulator described in Section 4, using dynamic obstacles and task appearance/disappearance to mimic disaster-zone uncertainty.

A standard scenario for swarm robotics search and rescue is shown in Figure 7. In this case, ground robots methodically coordinate their movements in the presence of local communications links and execute extensive surveys and debris field navigation. Drones in the air quickly make an examination of hazardous or inaccessible terrain, in order to detect potential survivors. Upon the detection of a survivor, the hybrid metaheuristic control approach reallocates ground units, dynamically re-organizing paths and tasks as obstacles and robotic failures present themselves. This combinatorial approach of both air and ground swarms under hybrid metaheuristic control provides quicker, more reliable, and scalable search and rescue.



**Figure 7.** Schematic of a swarm robotics search-and-rescue scenario

However, it should be noted that, even though the simulation results provided positive results, empirical validation in real-world with physical robots in field experiments will be essential to address the robustness of any detection systems under sensor noise, unsteady topography and noise caused by communication interference. Although the results of the simulations are promising, the scenarios they describe now rely on advanced but simplified models. The latest work involves validation of dynamic implementations and their physical testing.

## 8. CONCLUSIONS AND FUTURE WORK

Here we provide extensive and accurate evaluation of complex metaheuristic algorithms in swarm robotics, focusing specifically on a hybrid PSO–GA framework. The hybrid approach exhibited superior performance in task allocation, area coverage, robustness to failures, and scalability across a diverse range of scenarios, as demonstrated by comprehensive simulations and statistical analyses. By carrying out robust statistical comparisons with baseline algorithms, it was evident that these enhancements were paramount.

The main contribution of this paper is the demonstrated improvement in decentralized swarm task allocation and coverage through a hybrid PSO–GA controller, strengthened by distributed monocular ORB-SLAM2 for collaborative localization and mapping in dynamic, GPS-denied environments.

Future work will be directed toward the following paths:

- Conducting physical robot experiments: Evaluating hybrid metaheuristic algorithms on actual robotic platforms (e.g., TurtleBot, Kilobot) to assess practical performance and hardware limitations.
- Multi-objective, real-time optimization: Developing frameworks for the simultaneous enhancement of diverse objectives within dynamic, unpredictable, and three-dimensional environments.
- Foundational ethics and operational standards: Formulating precise protocols and ethical frameworks for the secure deployment of autonomous swarm robotic systems.

## REFERENCES

- [1] Al-Arbo, A.A., Al-Arbo, Y. (2026). Adaptive random-key particle swarm optimization with DC-closure local search for a two-stage fixed-charge transportation benchmark. *International Journal of Intelligent Engineering and Systems*, 19(4): 899 <https://doi.org/10.22266/ijies2026.0430.51>
- [2] Brambilla, M., Ferrante, E., Birattari, M., Dorigo, M. (2013). Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7: 1-41. <https://doi.org/10.1007/s11721-012-0075-2>
- [3] Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Springer Cham. <https://doi.org/10.1007/978-3-319-74528-2>
- [4] Dorigo, M., Birattari, M. (2007). Swarm intelligence. *Scholarpedia*, 2(9): 1462. <https://doi.org/10.4249/scholarpedia.1462>
- [5] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, Perth, WA, Australia, pp. 1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
- [6] Dias, M.B., Zlot, R., Kalra, N., Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7): 1257-1270. <https://doi.org/10.1109/JPROC.2006.876939>
- [7] Gerkey, B.P., Mataric, M.J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9): 939-954. <https://doi.org/10.1177/0278364904045564>
- [8] Al-Arbo, Y., Mahmood, H.F., Alqassab, A. (2025). End-to-end license plate detection and recognition in Iraq using a detection transformer and OCR. *Mesopotamian Journal of Computer Science*, 2025: 329-340.
- [9] Mondada, F., Bonani, M., Raemy, X., Pugh, J., et al. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, pp. 59-65. <https://infoscience.epfl.ch/entities/publication/d8bfd306-faf0-4dd3-9bf8-c00e455ade97>.
- [10] Rubenstein, M., Cornejo, A., Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198): 795-799. <https://doi.org/10.1126/science.1254295>
- [11] Fuller, A., Fan, Z., Day, C., Barlow, C. (2020). Digital twin: Enabling technologies, challenges and open research. *IEEE Access*, 8: 108952-108971. <https://doi.org/10.1109/ACCESS.2020.2998358>
- [12] Nguyen, T.T., Nguyen, N.D., Nahavandi, S. (2020). Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Transactions on Cybernetics*, 50(9): 3826-3839. <https://doi.org/10.1109/TCYB.2020.2977374>
- [13] Yang, X.S. (2010). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.
- [14] Mur-Artal, R., Tardós, J.D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5): 1255-1262. <https://doi.org/10.1109/TRO.2017.2705103>
- [15] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6): 1309-1332. <https://doi.org/10.1109/TRO.2016.2624754>
- [16] Zou, D.P., Tan, P., Yu, W.X. (2019). Collaborative visual SLAM for multiple agents: A brief survey. *Virtual Reality & Intelligent Hardware*, 1(5): 461-482. <https://doi.org/10.1016/j.vrih.2019.09.002>
- [17] Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press. <https://doi.org/10.7551/mitpress/1090.001.0001>
- [18] Khaleel, R.Z., Khaleel, H.Z., Al-Hareeri, A.A.A., Al-Obaidi, A.S.M., Humaidi, A.J. (2024). Improved trajectory planning of mobile robot based on pelican optimization algorithm. *Journal Européen des Systèmes Automatisés*, 57(4): 1005-1013. <https://doi.org/10.18280/jesa.570408>
- [19] Almola, O.S., Alkawaz, M.H., Al-Arbo, Y., Mahmood, H.F. (2025). Software-defined networking-enhanced fish-eye state routing framework for mobile wireless sensor networks. *Ingenierie des Systemes d'Information*, 30(11): 3037. <https://doi.org/10.18280/isi.301121>
- [20] Al-Arbo, A.A., Al-Kawaz, R.Z., Jameel, M.S. (2025). New meta-heuristic computer-oriented algorithms to solve unconstrained optimization problems. *Mathematical Modelling of Engineering Problems*, 12(3): 1081-1089. <https://doi.org/10.18280/mmep.120335>