AppliedMath

Review

Swarm Intelligence-Based Multi-Robotics: A Comprehensive Review

Luong Vuong Nguyen

Department of Artificial Intelligence, FPT University, Danang 550000, Vietnam; vuongnl3@fe.edu.vn

Abstract: Swarm Intelligence (SI) represents a paradigm shift in artificial intelligence, leveraging the collective behavior of decentralized, self-organized systems to solve complex problems. This study provides a comprehensive review of SI, focusing on its application to multi-robot systems. We explore foundational concepts, diverse SI algorithms, and their practical implementations by synthesizing insights from various reputable sources. The review highlights how principles derived from natural swarms, such as those of ants, bees, and birds, can be harnessed to enhance the efficiency, robustness, and scalability of multi-robot systems. We explore key advancements, ongoing challenges, and potential future directions. Through this extensive examination, we aim to provide a foundational understanding and a detailed taxonomy of SI research, paving the way for further innovation and development in theoretical and applied contexts.

Keywords: swarm intelligence; multi-robotics; swarm robots

check for updates

Citation: Nguyen, L.V. Swarm Intelligence-Based Multi-Robotics: A Comprehensive Review. *AppliedMath* **2024**, *4*, 1192–1210. https://doi.org/10.3390/appliedmath4040064

Academic Editors: Chengxi Zhang, Weisong Wen and Jin Wu

Received: 31 July 2024
Revised: 23 September 2024
Accepted: 29 September 2024
Published: 2 October 2024

1. Introduction

Swarm Intelligence (SI) is a burgeoning field within artificial intelligence that draws inspiration from the collective behavior observed in social insects such as ants, bees, and termites [1–3]. When applied to multi-robotics, SI facilitates the development of systems where numerous simple robots collaborate to perform complex tasks, exhibiting behaviors far more sophisticated than those of individual robots [4,5]. This decentralized approach emphasizes self-organization, local interactions, and emergent behavior, allowing robotic swarms to adapt dynamically to their environment and accomplish tasks efficiently. The study of SI in multi-robotics aims to harness these natural principles to create systems that are not only efficient but also resilient and scalable.

The motivation behind exploring SI in multi-robotics stems from its potential to revolutionize various industries and applications [6–8]. Traditional robotics often rely on centralized control systems, which can be bottlenecked by single points of failure and limited scalability. In contrast, SI offers robust, scalable, and flexible solutions operating in diverse and dynamic environments. The ability of swarms to self-organize and perform collective tasks without central oversight makes them ideal for applications such as search and rescue missions, environmental monitoring, agricultural automation, space exploration, and warehouse management [9–12]. These systems can perform complex tasks autonomously, adapt to new challenges, and continue operation even if individual robots fail. This inherent robustness and adaptability present a significant advantage over traditional robotic systems.

The research question that guides this review is: "How effective is Swarm Intelligence in enabling multi-robotic systems to perform critical tasks, and what are the key challenges and limitations in such applications?"

The specific objectives of this review are as follows:

- To provide an overview of the foundational principles and algorithms of Swarm Intelligence, including Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and other relevant algorithms used in multi-robotic systems.

- To examine the application of SI in key domains, such as search and rescue, environmental monitoring, agriculture, and space exploration.
- To identify and discuss the primary challenges and risks of using SI in multi-robot critical applications, focusing on areas like real-time decision-making, communication failures, and emergent behavior unpredictability.
- To propose potential future research directions and strategies for mitigating the identified risks, thereby improving the reliability and effectiveness of SI in multi-robotic systems for critical tasks.

The key focus of this study is a comprehensive review of SI, with a particular emphasis on its application to multi-robotics. By collecting studies from reputable sources (IEEE, Springer, ACM, and so on), this research aims to thoroughly understand the current advancements, challenges, and future directions in swarm intelligence and its implementation in multi-robot systems. In particular, our methodology for conducting this review is outlined below.

- Search strategy: Collect studies from reputation sources, including IEEE Xplore, SpringerLink, ScienceDirect, and Google Scholar, using search terms such as "swarm intelligence", "multi-robotics", "robotic swarms", "SI algorithms", "decentralized control", and "multi-agent systems". The inclusion criteria focused on peer-reviewed journal articles, conference papers, and significant book chapters published within the last 15 years (2008–2023) to ensure coverage of foundational research and recent advancements. We excluded non-English articles, non-peer-reviewed content, and papers not explicitly addressing the intersection of SI and multi-robotics.
- Selection: An initial screening of titles and abstracts was conducted to eliminate irrelevant studies, reducing the pool from over 200 articles to approximately 90. The remaining articles then underwent a full-text review, during which each paper was assessed for relevance to the review's objectives, methodological rigor, and contribution to the field. Ultimately, 65 articles were selected for inclusion based on their depth of analysis, novelty, and relevance to the research question.
- Synthesis of findings: The findings from the thematic and comparative analyses were synthesized to provide a comprehensive overview of the current state of research in SI-based multi-robotic systems. This synthesis also highlighted gaps in the literature and potential areas for future research.

We provide a taxonomy of recent research in swarm intelligence, categorizing the significant contributions and areas of focus within the field. As shown in Table 1, we summarize these categories, offering a clear overview of the current state of swarm intelligence research. In recap, the contributions of this study to the field of SI in multi-robotics are described as follows.

- Providing a detailed overview of the fundamental principles and concepts underpinning SI, elucidating how self-organization, decentralization, emergent behavior, and local interactions drive the functionality of robotic swarms.
- This study explores a wide range of applications where SI can effectively deploy, demonstrating its versatility and potential impact across different sectors.
- Identifying and analyzing the primary challenges of implementing SI in multi-robotic systems, offering insights into current limitations and areas requiring further research.
- We discuss the inherent limitations of SI, providing a balanced view of its capabilities and constraints and suggesting possible solutions to overcome these barriers.

The remainder of this study is structured as follows: Section 2 provides the key concepts of SI, providing a foundational understanding of its principles. This section covers self-organization, decentralization, emergent behavior, and local interactions, detailing how these principles are applied in robotic systems. Section 3 discusses the diverse applications of SI in multi-robotics, illustrating its practical uses in various fields such as search and rescue, environmental monitoring, agriculture, space exploration, and warehouse management. Section 4 describes the challenges faced in implementing SI systems, analyzing issues such as scalability, robustness, communication, coordination, heterogeneity, energy efficiency, adaptability, and security. Section 5 explores the limitations of SI, addressing concerns about the limited capabilities of individual robots, algorithmic complexity, unpredictability, dependence on local information, interference, congestion, real-world implementation difficulties, and maintenance and scalability costs. Finally, Section 6 summarizes the findings and suggests future research directions.

Table 1. Taxonomy of Swarm Intelligence research.

Category	Description	References
Foundational Concepts	The fundamental principles and frameworks of Swarm Intelligence, including its definition, characteristics, and basic models.	[1,2,13]
Swarm Intelligence Algorithms	Comprehensive reviews and surveys of various Swarm Intelligence algorithms such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Cuckoo Search, Firefly Algorithm, Bat Algorithm, and Artificial Bee Colony Algorithm.	[14–25]
Feature Selection	Applying Swarm Intelligence algorithms for feature selection in data mining and machine learning enhances the efficiency and accuracy of models.	[26–28]
Anomaly Detection	Use of Swarm Intelligence in developing robust anomaly detection systems to identify outliers and unusual patterns in data.	[29]
IoT and Cyber-Physical Systems	Integration of Swarm Intelligence algorithms in IoT-based systems and cyber-physical systems to enhance performance, scalability, and adaptability.	[30,31]
Swarm Robotics	Development and deployment of swarm robotics for various applications such as search and rescue, agriculture, space exploration, and autonomous systems.	[4–6,9,32–47]
Environmental Applications	Using Swarm Intelligence in environmental monitoring, pollution control, and sustainable management of natural resources.	[48–50]
Task Allocation and Path Planning	Swarm Intelligence approaches for optimizing task allocation and path planning in multi-robot systems to achieve efficient collaboration and resource utilization.	[10,51,52,52–54]

2. Swarm Intelligence

SI is the collective behavior of decentralized, self-organized systems, typically natural or artificial, which can solve complex problems through cooperation and interaction among simple agents [2,13,26]. The primary attributes of SI systems include flexibility, robustness, and scalability, derived from the simple rules followed by individuals and the indirect communication mechanisms among them [15]. This emergent property allows swarms to adapt to dynamic environments and efficiently find optimal solutions to complex problems.

SI systems exhibit several key characteristics that distinguish them from traditional approaches [27,29,55]. Decentralization is fundamental, as SI systems operate without central control; each agent follows simple rules based on local information and interactions. Through self-organization, agents autonomously organize into coherent structures or behaviors via local interactions. This leads to emergence, where complex global behaviors arise from these simple local rules and interactions. SI systems are also marked by robustness, able to withstand individual failures without significantly impacting overall performance [30]. Their flexibility allows them to adapt to changes in the environment or task requirements. Additionally, scalability ensures that the system's performance and behavior remain effective as the number of agents increases. Various animal behaviors exemplify natural SI [14]. Ant colonies use pheromone trails to communicate and coordinate tasks such as foraging and nest building. This behavior has inspired the Ant Colony Optimization (ACO) algorithm [56], where artificial ants build solutions based on pheromone trails. Bird flocking involves maintaining a cohesive formation while avoiding collisions and predators through simple rules like alignment, separation, and cohesion, which inspire the Particle Swarm Optimization (PSO) [18] algorithm. Fish schooling exhibits coordinated movements that help fish evade predators and improve foraging efficiency. In bee foraging, bees communicate the location of food sources through the waggle dance, influencing the colony's foraging patterns, and this behavior has led to the development of the bee algorithm, optimizing search processes.

A real-world example of natural SI can be seen in deploying autonomous drones for agricultural monitoring [38]. Inspired by the behaviors of ant colonies and bee foraging, these drones use algorithms like ACO and the bee algorithm to cover large fields efficiently, identify areas needing attention, and optimize their flight paths for data collection. The drones communicate and coordinate with each other by sharing information about detected issues, such as pest infestations or soil moisture levels, much like ants laying pheromone trails or bees performing the waggle dance. This decentralized and self-organizing approach allows drones to adapt to dynamic field conditions and ensure comprehensive coverage without central control, enhancing the efficiency and effectiveness of agricultural management. SI represents a fascinating and effective approach to solving complex problems through simple, local interactions among agents [28]. SI algorithms, such as ACO [17], PSO [18], Artificial Bee Colony (ABC) [19], Firefly Algorithm (FA) [57], Cuckoo Search (CS) [24], and Bat Algorithm (BA) [25], have demonstrated significant success in various applications by drawing inspiration from nature. These algorithms capitalize on decentralization, self-organization, and emergence principles, offering robust, flexible, and scalable solutions. Table 2 compares these SI algorithms across different criteria, such as convergence speed, scalability, complexity, and suitability for dynamic environments. The details of these algorithms are discussed in the following section.

Table 2. Comparison of Swarm Intelligence algorithms.

Algorithm	Convergence Speed	Scalability	Complexity	Suitability for Dynamic Environments
PSO	Fast in early stages; may stagnate at local optima	Moderate; performance decreases with increased dimensionality	Low computational complexity; easy to implement	Moderate; adaptable but may require modifications for real-time applications
ACO	Moderate; effective in finding global optima	High; can handle large-scale problems but at the cost of increased computational time	High due to the need for maintaining pheromone trails	High; well-suited for dynamic and changing environments due to its distributed nature
ABC	Moderate; depends on the neighborhood search strategy	High; can be scaled but with increased complexity	Moderate complexity; balancing exploration and exploitation can be challenging	Moderate to High; adaptable to dynamic environments but may require fine-tuning
FA	Fast; good at avoiding local optima	High; can scale well with parallel implementation	Moderate; requires careful parameter tuning	Low to Moderate; sensitive to dynamic changes, requires adaptation strategies
CS	Fast; effective in both global and local search	High; performs well on large-scale optimization problems	Moderate to high; requires proper tuning of parameters like step size	Low to Moderate; not inherently designed for dynamic environments but can be adapted
BA	Fast; quick convergence due to echolocation and frequency adjustment	Moderate; performance can degrade with problem size	Moderate; involves complex mechanisms like frequency and loudness adjustments	Moderate; can adapt to dynamic environments with parameter tuning

- Convergence Speed: Describes how quickly the algorithm approaches an optimal or near-optimal solution. Algorithms that converge too quickly may risk getting stuck in local optima, while slower convergence can lead to more thorough exploration.
- Scalability: Refers to the algorithm's ability to maintain performance as the problem size or dimensionality increases.
- Complexity: Relates to the computational demands and ease of algorithm implementation. High complexity may require more resources and be challenging to implement, especially in real-time systems.

- Suitability for Dynamic Environments: Indicates how well the algorithm can adapt to changes in environmental or problem parameters. Algorithms well-suited for dynamic environments are typically more flexible and can adjust to changes without significant performance degradation.

2.1. Ant Colony Optimization (ACO)

ACO [17,56] is a probabilistic technique inspired by the foraging behavior of ants, which use pheromone trails to communicate and find the shortest paths to food sources. In ACO, artificial ants build solutions by moving through a parameter space and depositing virtual pheromones, influencing the probability of future ants selecting specific paths. This approach is particularly effective for solving combinatorial optimization problems such as the Travelling Salesman Problem (TSP) and is increasingly being applied to multi-robot systems. The ACO algorithm is described as follows:

1. The algorithm starts by initializing the pheromone trails and other parameters.

$$\tau_{ij}(0) = \tau_0 \tag{1}$$

   where $\tau_{ij}(0)$ is the initial pheromone level on edge ij, and τ_0 is a small positive constant.
2. Each ant k constructs a solution by moving from one node to another. The probability $p_{ij}^k(t)$ of moving from node i to node j at time t is given by:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \tag{2}$$

   where,
   - $\tau_{ij}(t)$ is the pheromone level on edge ij at time t.
   - $\eta_{ij} = \frac{1}{d_{ij}}$ is the heuristic information (inverse of the distance d_{ij}).
   - α and β are parameters that control the influence of pheromone and heuristic information, respectively.
   - $\mathcal{N}_i^k$ is the set of feasible nodes for ant k when at node i.
3. After all ants have constructed their solutions, the pheromone levels are updated. The pheromone on edge ij is updated as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \tag{3}$$

   where,
   - ρ is the evaporation rate ($0 < \rho < 1$).
   - $\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k$ is the amount of pheromone deposited by ant k on edge ij, with:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ uses edge } ij \text{ in its solution} \\ 0 & \text{otherwise} \end{cases}$$

   - Q is a constant.
   - L_k is the length of the tour constructed by ant k.

Consider a scenario where a team of robots must navigate a warehouse to collect and deliver items to specified locations. Each robot uses ACO to determine the shortest paths while avoiding collisions and dynamic obstacles. Firstly, robots initialize pheromone levels across the warehouse grid. Then, each robot selects paths based on pheromone levels and heuristic information (e.g., distance to target) following Equation (2). After completing their paths, robots update the pheromone levels based on the efficiency of their routes according to Equation (3). Robots refine their paths through iterations, improving the warehouse operation's efficiency. The decentralized nature of ACO ensures robustness and scalability as the system adapts to the number of robots and dynamic changes within the environment.

2.2. Particle Swarm Optimization (PSO)

PSO [16,18] is a computational method inspired by the social behavior of birds flocking or fish schooling. It is used to find optimal solutions by having a group of candidate solutions, called particles, move around in the search space according to simple mathematical rules. Each particle adjusts its position based on its own experience and the experience of neighboring particles, aiming to find the best solution. PSO is particularly effective for continuous optimization problems and has been adapted for use in various fields, including multi-robot systems.

1. A swarm of particles is initialized with random positions and velocities in the search space. Each particle i has a position vector $\mathbf{x}_i$ and a velocity vector $\mathbf{v}_i$.

$$\mathbf{x}_i(0) = \text{rand}(\mathbf{x}_{\text{min}}, \mathbf{x}_{\text{max}}) \tag{4}$$

$$\mathbf{v}_i(0) = \text{rand}(\mathbf{v}_{\text{min}}, \mathbf{v}_{\text{max}}) \tag{5}$$

2. The velocity of each particle is updated based on its own best-known position $\mathbf{p}_i$ and the global best-known position $\mathbf{g}$.

$$\mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 r_1(\mathbf{p}_i(t) - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{g}(t) - \mathbf{x}_i(t)) \tag{6}$$

   where:
   - w is the inertia weight.
   - c_1 and c_2 are acceleration coefficients.
   - r_1 and r_2 are random numbers uniformly distributed in $[0, 1]$.
   - $\mathbf{p}_i(t)$ is the best-known position of particle i.
   - $\mathbf{g}(t)$ is the best-known position of the entire swarm.

3. The position of each particle is updated based on its new velocity.

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{7}$$

4. After updating the positions, the algorithm updates each particle's and global best-known positions.

$$\text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{p}_i(t)) \text{ then } \mathbf{p}_i(t+1) = \mathbf{x}_i(t+1) \tag{8}$$

$$\text{if } f(\mathbf{x}_i(t+1)) < f(\mathbf{g}(t)) \text{ then } \mathbf{g}(t+1) = \mathbf{x}_i(t+1) \tag{9}$$

In multi-robot systems, PSO can be used for various optimization tasks such as path planning, formation control, and task allocation. The algorithm's ability to find optimal solutions through collaboration and adaptation makes it suitable for decentralized robotic systems.

- Path Planning [34,35]: Robots use PSO to find optimal paths from starting points to destinations. Each robot represents a particle in the swarm, exploring different paths and sharing information about the best-found paths. This enables robots to converge on the shortest or safest paths while avoiding obstacles and minimizing travel time.
- Formation Control [58]: PSO can maintain a specific formation among multiple robots. Each robot adjusts its position to achieve the desired formation by optimizing the distances between itself and neighboring robots. This ensures the robots move cohesively and maintain formation despite environmental disturbances or changes.
- Task Allocation [36,37]: PSO can optimize the assignment of tasks to multiple robots, ensuring efficient use of resources and balanced workloads. Each particle represents a possible assignment of functions, and the swarm collaborates to find the optimal allocation. This allows the system to dynamically allocate tasks based on current conditions and robot capabilities.

2.3. Firefly Algorithm (FA)

FA [57] is a nature-inspired optimization algorithm based on the flashing behavior of fireflies. Fireflies use their bioluminescent flashes to attract mates and prey, and this behavior can be modeled to solve optimization problems. In FA, each firefly represents a potential solution, and its attractiveness is determined by its brightness, which is associated with the quality of the solution [20]. Fireflies move towards brighter and more attractive fireflies, leading the swarm to converge on optimal solutions. FA has been applied to various optimization problems and is particularly useful in multi-robot systems for path planning, task allocation, and formation control.

1. Its brightness determines a firefly's attractiveness β and decreases with distance. The attractiveness β_{ij} of firefly i to firefly j is given by:

$$\beta_{ij} = \beta_0 e^{-\gamma r_{ij}^2} \tag{10}$$

where:

- β_0 is the initial attractiveness.
- γ is the light absorption coefficient.
- r_{ij} is the distance between firefly i and firefly j.

2. The distance r_{ij} between firefly i at position $\mathbf{x}_i$ and firefly j at position $\mathbf{x}_j$ is calculated using the Euclidean distance:

$$r_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| \tag{11}$$

3. Firefly i moves towards a brighter firefly j according to:

$$\mathbf{x}_i = \mathbf{x}_i + \beta_{ij}(\mathbf{x}_j - \mathbf{x}_i) + \alpha(\text{rand} - 0.5) \tag{12}$$

where:

- α is a randomization parameter.
- rand is a uniformly distributed random number in [0,1].

4. The brightness I of a firefly is associated with the objective function value. For a minimization problem, the brightness I_i of firefly i can be inversely related to the objective function $f(\mathbf{x}_i)$:

$$I_i \propto \frac{1}{f(\mathbf{x}_i)} \tag{13}$$

Consider a scenario where a team of robots must navigate a cluttered environment to reach their respective goals. Each robot uses the Firefly Algorithm to find the optimal path while avoiding obstacles. Each robot (firefly) starts with a random initial position and evaluates its initial path's quality (brightness). Robots calculate the attractiveness of other robots' paths based on their brightness and the distance between them based on Equations (10) and (11). Then, robots move towards more attractive paths discovered by other robots. This can be formulated according to Equation (2). Finally, following Equation (13), robots update their paths and recalculate the brightness based on the new path quality. The robots refine their paths through iterations, converging on optimal paths that avoid obstacles and minimize travel time. The FA's decentralized nature ensures robustness and adaptability in dynamic environments.

2.4. Cuckoo Search (CS)

The CS [21,24] algorithm is a nature-inspired optimization algorithm. It is inspired by the brood parasitism behavior of some cuckoo species that lay their eggs in the nests of other host birds. The CS algorithm leverages this behavior to find optimal solutions to complex problems [22]. In CS, each potential solution is represented by a cuckoo's egg, and the goal is to replace the worst solutions with new, potentially better ones. Below are detailed step-by-step instructions for the CS algorithm process.

1. The position of a cuckoo is updated using a random walk called Levy flight, a step characterized by a heavy-tailed probability distribution. The new position $\mathbf{x}_i^{(t+1)}$ of cuckoo i is given by:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \cdot \text{Levy}(\lambda) \tag{14}$$

   where:
   - α is a step size scaling factor.
   - $\text{Levy}(\lambda)$ is a Levy flight distribution.
2. The Levy flight distribution is defined as:

$$\text{Levy}(\lambda) = t^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{15}$$

3. The fitness $f(\mathbf{x}_i)$ of each solution $\mathbf{x}_i$ is evaluated using the objective function of the optimization problem.
4. A fraction p_a of the worst nests (solutions) are replaced with new solutions:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha \cdot \text{Levy}(\lambda) \quad \text{if} \quad \text{rand} < p_a \tag{16}$$

   where p_a is the probability of discovering an alien egg.
5. The best solutions are carried over to the next generation based on their fitness values.

Consider a scenario where a team of robots must navigate an environment with obstacles to reach their respective goals. Each robot uses cuckoo search to find the optimal path. Firstly, each robot (cuckoo) starts with a random initial path and evaluates its fitness based on path length and obstacle avoidance. Robots update their paths using Levy flights to explore new potential paths as formulated in Equation (15). Then, they evaluate the new paths' fitness based on the objective function (e.g., path length and safety). Next, robots replace a fraction of the worst paths with new paths found through Levy flights if they provide better fitness (Equation (16)). The best paths are carried over to the next generation. The robots refine their paths through iterations, converging on optimal solutions that minimize travel time and avoid obstacles. The decentralized nature of CS ensures robustness and adaptability in dynamic environments.

2.5. Bat Algorithm (BA)

BA [25] is a nature-inspired optimization algorithm inspired by the echolocation behavior of bats. Bats use echolocation to navigate and locate prey in the dark, emitting sound waves and listening for the echoes that bounce back from obstacles. The BA leverages this behavior to find optimal solutions by mimicking the bats' ability to adjust their position and velocity based on the feedback from their environment [23]. The steps of BA are as follows:

1. Initialize the population of bats (solutions) $\mathbf{x}_i$ and their velocities $\mathbf{v}_i$. Assign random values to pulse frequency f_i, pulse rate r_i, and loudness A_i.
2. Update the frequency f_i, velocity $\mathbf{v}_i$, and position $\mathbf{x}_i$ of each bat:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{17}$$

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} + (\mathbf{x}_i^{(t)} - \mathbf{x}_{\text{best}})f_i \tag{18}$$

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)} \tag{19}$$

where:

- f_i is the frequency of the bat.
- $\beta \in [0, 1]$ is a random number.
- $\mathbf{x}_{\text{best}}$ is the current global best solution.

3. Generate a local solution around the best solution if a random number r is greater than the pulse rate r_i:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{best}} + \epsilon A_i^{(t)} \tag{20}$$

where ϵ is a random number drawn from a uniform distribution.

4. Update the loudness A_i and pulse rate r_i of each bat:

$$A_i^{(t+1)} = \alpha A_i^{(t)} \tag{21}$$

$$r_i^{(t+1)} = r_i^0[1 - \exp(-\gamma t)] \tag{22}$$

where α and γ are constants.

5. Accept the new solutions if they improve the objective function and update the best solution.

3. Applications of Swarm Intelligence in Multi-Robotics

SI has found numerous applications in multi-robotics across various fields [7]. By leveraging the collective behavior of simple agents, SI enables the development of robust, scalable, and efficient robotic systems capable of performing complex tasks. Below are detailed applications of SI in multi-robotics. Firstly, for consistency in understanding, we annotate all variables used in the remainder of this section in Table 3.

Table 3. List of variables used in the problem formulation for environmental monitoring, agricultural applications, and space exploration using SI.

Variable	Description
N	Number of drones/robots in the swarm
A	Total area of the forest/agricultural field/exploration zone
P_i	Position of the i-th drone/robot at time t
D_{ij}	Distance between drones/robots i and j
Q_i	Air quality measurement by the i-th drone (environmental monitoring)
T_i	Temperature measurement by the i-th drone (environmental monitoring)/time taken by the i-th robot to complete its task (agriculture, space)
H_i	Humidity measurement by the i-th drone (environmental monitoring)
V_i	Vegetation health index measured by the i-th drone (environmental monitoring)
R_i	Resources discovered by the i-th robot (space exploration)
C_i	Coverage area of the i-th drone (environmental monitoring)/crop health measurement by the i-th robot (agriculture)/communication signal strength of the i-th robot (space exploration)
E_i	Battery/energy level of the i-th drone/robot
F_i	Field area covered by the i-th robot (agriculture, space)
O_i	Penalty function for overlapping coverage
$f_{\min}, f_{\max}$	Minimum and maximum frequency values in bat algorithm
$\mathbf{v}_i$	Velocity of the i-th drone/robot
$\mathbf{x}_i$	Position vector of the i-th drone/robot
f_i	Frequency of the i-th drone/robot
r_i	Pulse rate of the i-th drone/robot
A_i	Loudness of the i-th drone/robot
$\alpha, \beta, \gamma, \delta, \epsilon$	Weights for the respective terms in the objective function

3.1. Search and Rescue

SI has found significant applications in multi-robot systems [32], particularly in the field of search and rescue (SAR) operations [51,52]. SAR missions often involve navigating complex and dynamic environments to locate and assist distressed individuals [53,54]. The decentralized and self-organizing nature of SI makes it well-suited for these tasks. Various SI algorithms, such as ACO, PSO, and BA, have been effectively applied in this context.

- Decentralized Coordination [52]: In SAR operations, multiple robots must explore large, often unknown, environments to find victims. Traditional centralized control approaches can be inefficient and prone to single points of failure. SI leverages decentralized coordination, where each robot operates based on local information and simple rules. This enhances robustness and scalability.
- Path Planning and Exploration [54]: SI algorithms excel in path planning and exploration, critical aspects of SAR missions. For instance, (i) ACO can simulate the behavior of ants foraging for food. Robots lay down virtual pheromone trails to mark paths that lead to successful discoveries (e.g., locating a victim). Other robots follow and reinforce these trails, optimizing the search paths over time; (ii) PSO can model the collective movement of birds or fish. Each robot adjusts its trajectory based on its own experience and the successes of its neighbors. This collaborative behavior helps cover more areas efficiently and quickly identify victims.

3.2. Environmental Monitoring

Swarm robotics impacts environmental monitoring [50] by collecting and analyzing data about the natural environment [48,49]. As shown in Figure 1, the IoT devices were used to monitor agriculture scenarios. This is essential to understanding and managing resources, detecting pollution, and monitoring ecological changes. The decentralized and adaptive nature of SI algorithms enhances the capabilities of multi-robot systems to perform these tasks efficiently and effectively. Applied swarm robotics, described below, can handle several issues during environmental monitoring.

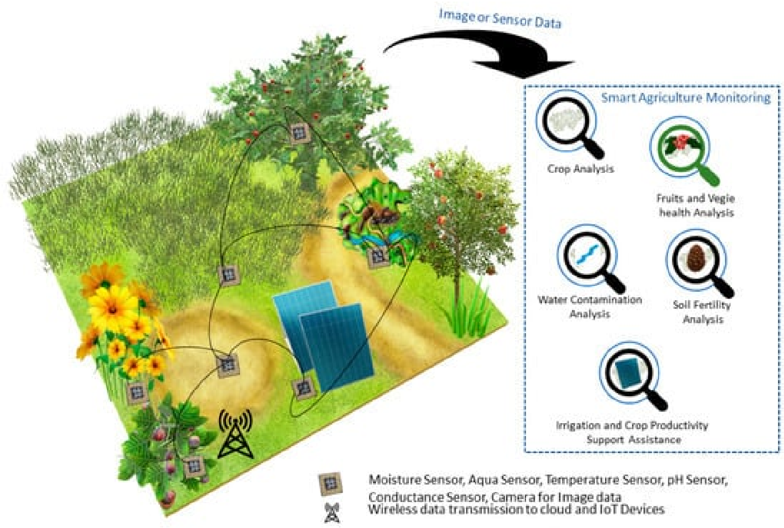

Figure 1. Smart agriculture monitoring system using IoT devices and sensors [50].

3.2.1. Area Coverage and Exploration

Effective environmental monitoring requires thorough and systematic coverage of the target area. SI algorithms can optimize the exploration and coverage tasks:

- ACO can be used where robots lay virtual pheromone trails to mark areas that have been explored. Other robots follow these trails to ensure thorough coverage without redundant overlaps.

- PSO can model the movement of robots as particles in a swarm, allowing them to cover large areas by adjusting their trajectories based on their own experiences and those of their neighbors.

3.2.2. Adaptability to Dynamic Environments

Environmental conditions can change rapidly, requiring the monitoring system to adapt [31]:

- BA principles can be used where robots adjust their paths using simulated echolocation. This helps them navigate dynamic and challenging environments like forests, wetlands, and underwater ecosystems.
- The flexibility of SI allows robots to respond in real time to changes in the environment, such as shifting weather conditions, water currents, or the presence of obstacles.

3.2.3. Data Fusion and Analysis

Once data are collected, they need to be aggregated and analyzed to draw meaningful conclusions [9]:

- Bee foraging behavior can inspire algorithms for data aggregation where robots communicate and share their findings, ensuring that data from different parts of the environment are combined efficiently.
- SI algorithms facilitate dynamic clustering and analysis of data, helping to identify patterns, anomalies, and trends in environmental parameters such as temperature, humidity, pollutant levels, and biodiversity.

3.2.4. Example

Consider a scenario where the quality of our environment is at stake. A forest area, a crucial part of our ecosystem, must be monitored for air quality, temperature, humidity, and vegetation health. The environment is dynamic, with potential obstacles like trees and varying terrain. A swarm of drones will be deployed to collect data and analyze environmental conditions, highlighting the significance of the task. There are the following constraints:

- Area Coverage: The entire forest area must be covered without significant overlaps or gaps.
- Obstacle Avoidance: Drones must navigate around trees, branches, and other obstacles.
- Battery Life: Each drone has limited battery life, requiring efficient path planning.
- Data Reliability: Data must be accurate and representative of the monitored parameters.

The objective is to maximize the coverage and accuracy of environmental data collection while minimizing energy consumption and ensuring robustness. The objective function F can be formulated as follows.

$$F = \alpha\left(\frac{\sum_{i=1}^{N} C_i}{A}\right) + \beta\left(\frac{\sum_{i=1}^{N}(Q_i + T_i + H_i + V_i)}{N}\right) - \gamma\left(\frac{\sum_{i=1}^{N} E_i}{N}\right) - \delta\left(\frac{\sum_{i=1}^{N} O_i}{N}\right) \quad (23)$$

where all variables were annotated in Table 3.

3.3. Agriculture

SI offers significant potential for transforming agricultural practices by deploying multi-robot systems [38,59]. By leveraging the collective behavior of simple agents, SI enables efficient, scalable, and adaptive solutions to complex agricultural challenges [40,41]. This section discusses several applications of SI in multi-robotics for agriculture, highlighting their benefits and impact.

- Field Coverage and Monitoring [60,61]: In large agricultural fields, uniform coverage and monitoring are critical for soil analysis, crop health assessment, and pest detection. Multi-robot systems, guided by SI algorithms, can ensure comprehensive field coverage by distributing the robots evenly across the area.

- Planting and Seeding [39]: Precision planting and seeding are crucial for maximizing crop yield and minimizing waste. SI algorithms enable robots to coordinate their movements and actions to plant seeds at optimal distances and depths.
- Pest and Weed Control [9]: Managing pests and weeds is vital for healthy crop growth. Swarm robotics can precisely deploy robots to identify and target pests and weeds, reducing the need for blanket pesticide applications.
- Harvesting [62]: Harvesting crops is labor-intensive and time-sensitive. Swarm robots can collaborate to harvest crops efficiently, ensuring minimal damage and maximizing yield.

To design a multi-robot system using SI algorithms for efficient agricultural tasks, the primary goals are to ensure optimal field coverage, effective pest control, efficient planting and harvesting, and real-time adaptability to environmental changes. For example, a large agricultural field needs to be monitored and maintained. The tasks include planting seeds, applying pesticides, monitoring crop health, and harvesting crops. The environment is dynamic, with varying weather conditions and possible obstacles like irrigation systems. A swarm of robots will be deployed to perform these tasks efficiently.

- Field Coverage: The agricultural field must be covered without significant overlaps or gaps.
- Obstacle Avoidance: Robots must navigate around irrigation systems, fences, and other obstacles.
- Battery Life: Each robot has limited battery life, requiring efficient path planning.
- Task Efficiency: Robots must perform tasks such as planting, spraying, and harvesting efficiently.
- Data Reliability: Data regarding crop health and environmental conditions must be accurate.

The objective is to maximize the coverage and efficiency of agricultural tasks while minimizing energy consumption and ensuring robust performance. The objective function F can be formulated as:

$$F = \alpha\left(\frac{\sum_{i=1}^{N} F_i}{A}\right) + \beta\left(\frac{\sum_{i=1}^{N} C_i}{N}\right) - \gamma\left(\frac{\sum_{i=1}^{N} T_i}{N}\right) - \delta\left(\frac{\sum_{i=1}^{N} E_i}{N}\right) - \epsilon\left(\frac{\sum_{i=1}^{N} O_i}{N}\right) \tag{24}$$

where all variables are shown in Table 3.

3.4. Space Exploration

The collective behavior of simple robotic agents in SI systems enables robust, scalable, and efficient operations in space's harsh and dynamic environment [42,43]. This section provides several key applications of SI in multi-robotics for space exploration, highlighting their benefits and potential impacts.

3.4.1. Planetary Exploration

- Surface Mapping and Analysis [44,45]: Swarm robots can be deployed on planetary surfaces to map and analyze the terrain collaboratively. Each robot in the swarm can be equipped with different sensors to collect data on soil composition, topography, and environmental conditions. The benefits of this application are that it ensures comprehensive and high-resolution mapping, reduces the risk of mission failure by distributing tasks among multiple robots, and enhances data accuracy through collective measurements.
- Resource Discovery and Utilization [46]: A swarm robot can explore planetary surfaces to locate and assess resources such as water, minerals, and other materials. This information is crucial for in situ resource utilization (ISRU), which aims to use local resources for mission sustainability. The benefits are that it increases the efficiency of resource discovery, reduces the need for transporting materials from Earth, and supports long-term human and robotic presence on other planets.

3.4.2. Swarm-Based Construction

- Habitat Construction [31]: Swarm robots can construct habitats and other structures on planetary surfaces. By coordinating their actions, the robots can build complex structures using local materials or pre-fabricated components. Benefits: Enhances construction efficiency, reduces the need for human labor, and ensures precision and robustness in habitat construction.
- Infrastructure Deployment [33]: Robots can deploy and assemble infrastructure such as solar panels, communication arrays, and scientific instruments. The benefits are that it enables rapid deployment of essential infrastructure, supports the establishment of permanent bases, and ensures redundancy and resilience through distributed systems.

3.4.3. In-Space Assembly and Maintenance

- Satellite Swarming [47]: Swarms of small satellites can work together to perform tasks such as Earth observation, communication, and space weather monitoring. These satellite swarms can reconfigure themselves in orbit to optimize performance. Benefits: Increases the flexibility and coverage of satellite networks, reduces the cost of individual satellites, and enhances the resilience of satellite systems.
- Spacecraft Maintenance and Repair [63]: Swarm robots can perform maintenance and repair tasks on spacecraft, space stations, and other space assets. They can inspect structures, identify issues, and perform repairs autonomously. Benefits: Reduces the need for risky extravehicular activities (EVAs) by astronauts, ensures continuous operation of space assets, and extends the lifespan of spacecraft.

3.4.4. Example

The primary goal is to design a swarm robotic system to efficiently and autonomously explore planetary surfaces. The primary goals are ensuring comprehensive terrain mapping, resource discovery, and infrastructure deployment while maintaining energy efficiency and reliability. A swarm of robots is deployed on a planetary surface to perform tasks such as terrain mapping, resource discovery, and infrastructure construction. The environment is unknown and dynamic, with potential obstacles and varying environmental conditions.

- Field Coverage: The exploration area must be covered without significant overlaps or gaps.
- Obstacle Avoidance: Robots must navigate around rocks, craters, and other obstacles.
- Energy Efficiency: Each robot has limited energy resources, requiring efficient path planning and task allocation.
- Communication: Reliable communication must be maintained among the robots and the base station.
- Data Accuracy: Data collected on terrain, resources, and environmental conditions must be accurate and reliable.

The objective is to maximize exploration coverage and resource discovery while minimizing energy consumption and ensuring robust communication. The objective function F can be formulated as:

$$F = \alpha\left(\frac{1}{A}\sum_{i=1}^{N} F_i\right) + \beta\left(\frac{1}{N}\sum_{i=1}^{N} R_i\right) + \epsilon\left(\frac{1}{N}\sum_{i=1}^{N} C_i\right) - \delta\left(\frac{1}{N}\sum_{i=1}^{N} E_i\right) - \gamma\left(\frac{1}{N}\sum_{i=1}^{N} T_i\right) \quad (25)$$

where all variables are shown in Table 3.

4. Challenges in Implementing Swarm Intelligence-Based Multi-Robotics

Implementing SI in multi-robotic systems presents various challenges [59]. These challenges arise from the need to coordinate numerous robots operating in dynamic, often unpredictable environments without centralized control. The following sections detail these challenges:

- Scalability: Scalability is a critical challenge in swarm robotics. Maintaining efficiency becomes more complex as the number of robots in a swarm increases. The system must ensure that each robot contributes to the objective without causing redundant efforts or bottlenecks. The computational and communication overhead can increase significantly, necessitating efficient algorithms that scale without compromising performance.
  Consider a case study in a large-scale multi-robotic exploration project. Scalability was a significant issue due to the increased number of robots and the complexity of coordinating their activities. The project employed hierarchical control architectures and clustering techniques to manage communication and processing loads. The hierarchical control structure allowed the system to delegate decision-making to cluster heads, reducing the need for constant communication with a central controller. Clustering robots based on geographic location or assigned tasks minimized communication overhead, enhancing the system's scalability without overwhelming it.
- Robustness: Robustness refers to the swarm's ability to handle individual robot failures without degrading overall performance. In a large swarm, some robots will likely experience mechanical failures, power shortages, or environmental damage. The system must be designed to compensate for these losses dynamically, redistributing tasks among the remaining robots and ensuring the mission objectives are still met.
  In disaster response scenarios, robots are often deployed in hazardous environments where individual units are prone to failure. Robustness is critical, as the failure of one or more robots should not compromise the overall mission. A practical approach is using fault-tolerant algorithms that allow for dynamic task reallocation and reconfiguration of the swarm. For instance, when a robot fails during a search-and-rescue operation, nearby robots automatically adjust their search patterns to cover the area initially assigned to the failed unit, ensuring continuous coverage.
- Communication: Ensuring reliable information exchange among robots is another major challenge. Maintaining stable communication links can be difficult in many environments, particularly with obstacles or harsh conditions. Communication protocols must be robust against packet loss, interference, and delays and facilitate efficient data sharing without overwhelming the network bandwidth.
- Coordination: Achieving coordinated actions among many robots without central control requires sophisticated algorithms. These algorithms must enable robots to synchronize activities, avoid conflicts, and work towards common goals. Coordination mechanisms often rely on local interactions and decentralized decision-making, which can be complex to design and implement effectively.
- Heterogeneity: In many applications, swarms may consist of heterogeneous robots with different capabilities, sensors, and power levels. Integrating these diverse robots into a cohesive system that can operate efficiently is challenging. The control algorithms must account for these differences, ensuring each robot contributes optimally according to its strengths and limitations.
  Coordination is essential for formation control and obstacle avoidance in autonomous vehicle swarms. A notable study developed a decentralized coordination algorithm based on local interactions between vehicles. Each vehicle made decisions based on its immediate neighbors, allowing the swarm to maintain a cohesive formation and avoid collisions. This approach highlights the effectiveness of decentralized coordination in achieving complex group behaviors without centralized control.
- Energy Efficiency: Energy efficiency is crucial for the extended operation of swarm robotic systems, especially in remote or harsh environments where recharging may not be feasible. Minimizing energy consumption includes optimizing path planning, task allocation, and communication protocols. The system must balance the energy expenditure with the mission requirements to ensure longevity and reliability.
- Adaptability: Swarm robotic systems must be adaptable to dynamic environments. This involves adjusting to changes in the terrain, obstacles, environmental conditions,

and unexpected events. The algorithms must enable real-time responses to new information, allowing the swarm to modify its behavior and flight strategies.
In a search-and-rescue mission following a natural disaster, the environment can change rapidly due to shifting debris, weather conditions, or other factors. An adaptable swarm of robots was designed to modify its search patterns in real time based on sensor data. When an area became inaccessible due to new obstacles, the robots quickly adjusted their routes and continued searching other regions, demonstrating the importance of adaptability in dynamic environments.
- Security: Security is important in swarm robotics, particularly for sensitive or hostile environment applications. Protecting the swarm from malicious attacks, such as hacking or the introduction of rogue robots, is essential. Security measures include robust encryption of communication channels, authentication protocols, and the ability to detect and isolate compromised robots.
In military applications, ensuring the security of robotic swarms is vital to prevent adversaries from compromising the mission. A case study in this area involved implementing encrypted communication and multi-factor authentication for all robots in the swarm. Additionally, the system included anomaly detection algorithms that could identify and isolate any robot exhibiting suspicious behavior, such as deviating from its assigned tasks, thus maintaining the integrity of the swarm.

5. Limitations of Swarm Intelligence-Based Multi-Robotics

While swarm intelligence-based multi-robotic systems offer significant potential, they also face substantial limitations. These limitations can hinder their performance and deployment in various real-world applications. Addressing these limitations requires ongoing research and development. Below are some key limitations and the current trends and future work to address them:

- Limited Individual Capability: Swarm robots often have constraints on their sensing, processing, and action capabilities. This limited capability can restrict the types of tasks they can perform and the environments in which they can operate effectively. Regarding this limitation, current trends aim to enhance individual robot capabilities through advances in sensor technology, onboard processing power, and more efficient actuation systems. Future work includes integrating AI and machine learning to enable better decision-making and more sophisticated behaviors.
- Complexity of Algorithm Design: Developing reliable algorithms that produce the desired collective behaviors from simple individual rules is highly challenging. Ensuring these robust, efficient, and adaptable algorithms is a significant hurdle. For this issue, research focuses on bio-inspired algorithms, such as those mimicking ant foraging or bird flocking. Machine learning, particularly reinforcement learning, is being explored to develop algorithms to learn optimal behaviors. Future work includes creating more modular and reusable algorithmic frameworks.
- Unpredictability: Due to the complexity and non-linearity of interactions among robots, swarm systems can exhibit unpredictable behaviors. This unpredictability can lead to undesired emergent outcomes that are difficult to foresee and control. Efforts are being made to better understand and model emergent behaviors through simulations and theoretical analysis. Future research aims to develop control mechanisms to mitigate undesirable emergent behaviors and enhance predictability.
- Dependence on Local Information: Swarm robots typically rely on local information and interactions to make decisions, which can lead to suboptimal global performance if local data are insufficient or misleading. Enhancing the ability of robots to share information and make more informed decisions based on a combination of local and global data. Future work includes developing hybrid approaches that balance local autonomy with occasional global coordination.
- Interference and Congestion: High robot density can lead to interference and congestion, reducing the overall efficiency of the swarm. This is particularly problematic

in confined or resource-limited environments. Developing advanced path planning and collision avoidance algorithms to minimize interference. Future research explores dynamic density management strategies where robots can adapt their density based on task requirements and environmental conditions.

- Difficulty in Real-World Implementation: Translating theoretical models and simulations of swarm intelligence to practical, real-world systems is challenging. Environmental unpredictability, hardware limitations, and real-time constraints complicate implementation. Bridging the gap between theory and practice through extensive field testing and iterative development. Future work includes creating more robust and versatile robotic platforms that can operate reliably in diverse real-world conditions.
- Maintenance and Scalability Costs: Large-scale deployment of swarm robotic systems can be financially demanding due to the costs associated with manufacturing, maintenance, and scalability. This includes not only the initial investment but also ongoing operational expenses. Reducing costs through advances in manufacturing technologies, such as 3D printing and modular design. Developing maintenance strategies that leverage the swarm's inherent redundancy and fault tolerance. Future work includes exploring economic models that make large-scale deployment more viable.

Despite these limitations, SI has demonstrated remarkable success in various multi-robot applications, particularly in non-critical tasks like exploration, mapping, and collective transportation. However, using SI in critical applications (such as search and rescue, space exploration, or military operations) introduces several potential drawbacks and risks. These include communication failures, unpredictability, real-time decision-making challenges, and resource constraints. Addressing these drawbacks will require further advancements in control algorithms, communication technologies, and robust decision-making frameworks to ensure the reliability and safety of SI systems in critical, high-stakes environments.

6. Conclusions

Swarm intelligence in multi-robotics enables robots to perform complex tasks through simple local interactions without centralized control, offering advancements in search and rescue, environmental monitoring, agriculture, and space exploration. Despite challenges like limited capabilities and high maintenance costs, the field is rapidly progressing. Ongoing research improves scalability, robustness, communication, coordination, energy efficiency, and security. Innovations in sensor technology, machine learning, and distributed computing enhance swarm systems. Bio-inspired algorithms and reinforcement learning improve coordination, while advances in communication protocols and energy-efficient hardware boost reliability and longevity. These advancements bring swarm intelligence closer to real-world applications, offering efficient, resilient robotic systems for complex missions and scalable solutions for large-scale deployments. The future lies in developing robust, adaptable swarms for diverse environments, with advanced AI, improved hardware, and innovative algorithms further enhancing autonomy and effectiveness.

Funding: This research received no external funding

Data Availability Statement: This article includes the data presented in the study.

Acknowledgments: This research was supported by AIT Laboratory, FPT University, Danang Campus, Vietnam 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kennedy, J. Swarm intelligence. In *Handbook of Nature-Inspired and Innovative Computing: Integrating Classical Models with Emerging Technologies*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 187–219.
2. Beni, G. Swarm intelligence. In *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*; Springer: New York, NY, USA , 2020; pp. 791–818.

3. Wang, C.; Zhang, S.; Ma, T.; Xiao, Y.; Chen, M.Z.; Wang, L. Swarm intelligence: A survey of model classification and applications. *Chin. J. Aeronaut.* **2024**, *in press*. [CrossRef]
4. Dorigo, M.; Theraulaz, G.; Trianni, V. Swarm robotics: Past, present, and future [point of view]. *Proc. IEEE* **2021**, *109*, 1152–1165. [CrossRef]
5. Khaldi, B.; Cherif, F. An overview of swarm robotics: Swarm intelligence applied to multi-robotics. *Int. J. Comput. Appl.* **2015**, *126*, 31–37. [CrossRef]
6. Chung, S.J.; Paranjape, A.A.; Dames, P.; Shen, S.; Kumar, V. A survey on aerial swarm robotics. *IEEE Trans. Robot.* **2018**, *34*, 837–855. [CrossRef]
7. Cheraghi, A.R.; Shahzad, S.; Graffi, K. Past, present, and future of swarm robotics. In *Intelligent Systems and Applications, Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3, Amsterdam, The Netherlands, 2–3 September 2021*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 190–233.
8. Olaronke, I.; Rhoda, I.; Gambo, I.; Ojerinde, O.; Janet, O. A systematic review of swarm robots. *Curr. J. Appl. Sci. Technol.* **2020**, *39*, 79–97. [CrossRef]
9. Schranz, M.; Umlauft, M.; Sende, M.; Elmenreich, W. Swarm robotic behaviors and current applications. *Front. Robot. AI* **2020**, *7*, 36. [CrossRef]
10. Connor, J.; Champion, B.; Joordens, M.A. Current algorithms, communication methods and designs for underwater swarm robotics: A review. *IEEE Sens. J.* **2020**, *21*, 153–169. [CrossRef]
11. Ibrahim, R.; Alkilabi, M.; Khayeat, A.R.H.; Tuci, E. Review of collective decision making in swarm robotics. *J. Al-Qadisiyah Comput. Sci. Math.* **2024**, *16*, 72–80. [CrossRef]
12. Majid, M.; Arshad, M.; Mokhtar, R. Swarm robotics behaviors and tasks: A technical review. In *Control Engineering in Robotics and Industrial Automation: Malaysian Society for Automatic Control Engineers (MACE) Technical Series 2018*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 99–167.
13. Bonabeau, E. Swarm intelligence. In *A Primer on Multiple Intelligences*; Springer: Berlin/Heidelberg, Germany, 2003; p. 211.
14. Selvaraj, S.; Choi, E. Survey of swarm intelligence algorithms. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020; pp. 69–73.
15. Tang, J.; Liu, G.; Pan, Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1627–1643. [CrossRef]
16. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle swarm optimization: A comprehensive survey. *IEEE Access* **2022**, *10*, 10031–10061. [CrossRef]
17. Dorigo, M.; Socha, K. An introduction to ant colony optimization. In *Handbook of Approximation Algorithms and Metaheuristics*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 395–408.
18. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [CrossRef]
19. Kaya, E.; Gorkemli, B.; Akay, B.; Karaboga, D. A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105311. [CrossRef]
20. Kumar, V.; Kumar, D. A systematic review on firefly algorithm: Past, present, and future. *Arch. Comput. Methods Eng.* **2021**, *28*, 3269–3291. [CrossRef]
21. Guerrero-Luis, M.; Valdez, F.; Castillo, O. A review on the cuckoo search algorithm. In *Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 113–124.
22. Cuong-Le, T.; Minh, H.L.; Khatir, S.; Wahab, M.A.; Tran, M.T.; Mirjalili, S. A novel version of Cuckoo search algorithm for solving optimization problems. *Expert Syst. Appl.* **2021**, *186*, 115669. [CrossRef]
23. Li, G.; Xu, H.; Lin, Y. Application of bat algorithm based time optimal control in multi-robots formation reconfiguration. *J. Bionic Eng.* **2018**, *15*, 126–138. [CrossRef]
24. Mareli, M.; Twala, B. An adaptive Cuckoo search algorithm for optimisation. *Appl. Comput. Inform.* **2018**, *14*, 107–115. [CrossRef]
25. Cui, Z.; Li, F.; Zhang, W. Bat algorithm with principal component analysis. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 603–622. [CrossRef]
26. Rostami, M.; Berahmand, K.; Nasiri, E.; Forouzandeh, S. Review of swarm intelligence-based feature selection methods. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104210. [CrossRef]
27. Nguyen, B.H.; Xue, B.; Zhang, M. A survey on swarm intelligence approaches to feature selection in data mining. *Swarm Evol. Comput.* **2020**, *54*, 100663. [CrossRef]
28. Brezočnik, L.; Fister, I., Jr.; Podgorelec, V. Swarm intelligence algorithms for feature selection: A review. *Appl. Sci.* **2018**, *8*, 1521. [CrossRef]
29. Mishra, S.; Sagban, R.; Yakoob, A.; Gandhi, N. Swarm intelligence in anomaly detection systems: An overview. *Int. J. Comput. Appl.* **2021**, *43*, 109–118. [CrossRef]
30. Zedadra, O.; Guerrieri, A.; Jouandeau, N.; Spezzano, G.; Seridi, H.; Fortino, G. Swarm intelligence-based algorithms within IoT-based systems: A review. *J. Parallel Distrib. Comput.* **2018**, *122*, 173–187. [CrossRef]
31. Schranz, M.; di Caro, G.A.; Schmickl, T.; Elmenreich, W.; Arvin, F.; Şekercioğlu, A.; Sende, M. Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends. *Swarm Evol. Comput.* **2021**, *60*, 100762. [CrossRef]
32. Tan, Y. Swarm robotics: Collective behavior inspired by nature. *J. Comput. Sci. Syst. Biol.* **2013**, *6*, e106.

33. Dias, P.G.F.; Silva, M.C.; Rocha Filho, G.P.; Vargas, P.A.; Cota, L.P.; Pessin, G. Swarm robotics: A perspective on the latest reviewed concepts and applications. *Sensors* **2021**, *21*, 2062. [CrossRef]
34. Thabit, S.; Mohades, A. Multi-robot path planning based on multi-objective particle swarm optimization. *IEEE Access* **2018**, *7*, 2138–2147. [CrossRef]
35. Das, P.; Jena, P.K. Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators. *Appl. Soft Comput.* **2020**, *92*, 106312. [CrossRef]
36. Wei, C.; Ji, Z.; Cai, B. Particle Swarm Optimization for Cooperative Multi-Robot Task Allocation: A Multi-Objective Approach. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2530–2537. [CrossRef]
37. Kong, X.; Gao, Y.; Wang, T.; Liu, J.; Xu, W. Multi-robot task allocation strategy based on particle swarm optimization and greedy algorithm. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; pp. 1643–1646.
38. Albiero, D.; Garcia, A.P.; Umezu, C.K.; de Paulo, R.L. Swarm Robots in Agriculture. *arXiv* **2021**, arXiv:2103.06732.
39. Botteghi, N.; Kamilaris, A.; Sinai, L.; Sirmaçek, B. Multi-agent path planning of robotic swarms in agricultural fields. In Proceedings of the XXIVth ISPRS Congress 2020, Nice, France, 4–10 July 2020; pp. 361–368.
40. Lytridis, C.; Kaburlasos, V.G.; Pachidis, T.; Manios, M.; Vrochidou, E.; Kalampokas, T.; Chatzistamatis, S. An overview of cooperative robotics in agriculture. *Agronomy* **2021**, *11*, 1818. [CrossRef]
41. David, C.N.A.; Yumol, J.V.; Garcia, R.G.; Ballado, A.H. Swarm robotics application for gathering soil samples. In Proceedings of the 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Manila, Philippines, 28–30 November 2021; pp. 1–6.
42. Gul, F.; Mir, I.; Rahiman, W.; Islam, T.U. Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm. *IEEE Access* **2021**, *9*, 22774–22787. [CrossRef]
43. Gul, F.; Mir, I.; Abualigah, L.; Sumari, P. Multi-robot space exploration: An augmented arithmetic approach. *IEEE Access* **2021**, *9*, 107738–107750. [CrossRef]
44. Quattrini Li, A. Exploration and mapping with groups of robots: Recent trends. *Curr. Robot. Rep.* **2020**, *1*, 227–237. [CrossRef]
45. Lluvia, I.; Lazkano, E.; Ansuategi, A. Active mapping and robot exploration: A survey. *Sensors* **2021**, *21*, 2445. [CrossRef] [PubMed]
46. Thangavelautham, J. Autonomous robot swarms for off-world construction and resource mining. In Proceedings of the AIAA Scitech 2020 Forum, Orlando, FL, USA, 6–10 January 2020; p. 0795.
47. Foust, R.C.; Lupu, E.S.; Nakka, Y.K.; Chung, S.J.; Hadaegh, F.Y. Autonomous in-orbit satellite assembly from a modular heterogeneous swarm. *Acta Astronaut.* **2020**, *169*, 191–205. [CrossRef]
48. Xiang, X.; Li, Q.; Khan, S.; Khalaf, O.I. Urban water resource management for sustainable environment planning using artificial intelligence techniques. *Environ. Impact Assess. Rev.* **2021**, *86*, 106515. [CrossRef]
49. Ye, Z.; Yang, J.; Zhong, N.; Tu, X.; Jia, J.; Wang, J. Tackling environmental challenges in pollution controls using artificial intelligence: A review. *Sci. Total Environ.* **2020**, *699*, 134279. [CrossRef]
50. Ullo, S.L.; Sinha, G.R. Advances in smart environment monitoring systems using IoT and sensors. *Sensors* **2020**, *20*, 3113. [CrossRef]
51. Paez, D.; Romero, J.P.; Noriega, B.; Cardona, G.A.; Calderon, J.M. Distributed particle swarm optimization for multi-robot system in search and rescue operations. *IFAC-PapersOnLine* **2021**, *54*, 1–6. [CrossRef]
52. Horyna, J.; Baca, T.; Walter, V.; Albani, D.; Hert, D.; Ferrante, E.; Saska, M. Decentralized swarms of unmanned aerial vehicles for search and rescue operations without explicit communication. *Auton. Robot.* **2023**, *47*, 77–93. [CrossRef]
53. Ruetten, L.; Regis, P.A.; Feil-Seifer, D.; Sengupta, S. Area-optimized UAV swarm network for search and rescue operations. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 613–618.
54. Drew, D.S. Multi-agent systems for search and rescue applications. *Curr. Robot. Rep.* **2021**, *2*, 189–200. [CrossRef]
55. Wang, J.; Lin, Y.; Liu, R.; Fu, J. Odor source localization of multi-robots with swarm intelligence algorithms: A review. *Front. Neurorobotics* **2022**, *16*, 949888. [CrossRef] [PubMed]
56. Dorigo, M.; Stützle, T. *Ant Colony Optimization: Overview and Recent Advances*; Springer: Berlin/Heidelberg, Germany, 2019.
57. Yang, X.S.; Slowik, A. Firefly algorithm. In *Swarm Intelligence Algorithms*; CRC Press: Boca Raton, FL, USA, 2020; pp. 163–174.
58. Kamel, M.A.; Yu, X.; Zhang, Y. Real-time fault-tolerant formation control of multiple WMRs based on hybrid GA–PSO algorithm. *IEEE Trans. Autom. Sci. Eng.* **2020**, *18*, 1263–1276. [CrossRef]
59. Albiero, D.; Garcia, A.P.; Umezu, C.K.; de Paulo, R.L. Swarm robots in mechanized agricultural operations: A review about challenges for research. *Comput. Electron. Agric.* **2022**, *193*, 106608. [CrossRef]
60. Albani, D.; Manoni, T.; Arik, A.; Nardi, D.; Trianni, V. Field coverage for weed mapping: Toward experiments with a UAV swarm. In Proceedings of the Bio-Inspired Information and Communication Technologies: 11th EAI International Conference, BICT 2019, Proceedings 11, Pittsburgh, PA, USA, 13–14 March 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 132–146.
61. Marwah, N.; Singh, V.K.; Kashyap, G.S.; Wazir, S. An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning. *Int. J. Inf. Technol.* **2023**, *15*, 2317–2327. [CrossRef]

62. Pooranam, N.; Vignesh, T. A Swarm Robot for Harvesting a Paddy Field. In *Nature-Inspired Algorithms Applications*; Wiley: Hoboken, NJ, USA, 2021; pp. 137–156.
63. Asri, E.G.; Zhu, Z.H. An introductory review of swarm technology for spacecraft on-orbit servicing. *Int. J. Mech. Syst. Dyn.* **2024**, *4*, 3–21. [CrossRef]