Computer Science • 26(1) 2025 https://doi.org/10.7494/csci.2025.26.1.6306

Rachid Adrdor

THE POWER OF INTELLIGENCE EMERGING FROM SWARMS

Abstract *Swarm intelligence (SI) is a field of study that seeks to understand and model collective behaviors observed in natural social systems, such as ant colonies, bee hives, termite mounds, flocks of birds or schools of fish. The central principle of SI is that complex intelligent behaviors can emerge from the interactions of large numbers of simple individual entities, without any centralized control or monitoring. SI researchers aim to uncover the underlying principles and mechanisms behind this SI, with the aim of applying these concepts to solve complex problems in areas such as optimization, robotics, transport, IT, etc. As the field continues to evolve, SI is expected to have an increasingly significant impact on our understanding of biological systems and our ability to design intelligent systems capable of adapting and thriving in complex environments and dynamic. This article aims to introduce the reader to the field of SI, presenting its fundamental concepts, key principles, existing applications, and prospective future developments.*

Keywords swarm intelligence, collective behavior, optimization

Citation Computer Science 26(1) 2025: 77–100

1. Introduction

Nowadays, there is increasing interest in solving optimization problems without relying on prior knowledge. Indeed, prior knowledge about the problem, such as the mathematical model, constraints, and objective function, is often necessary to design effective optimization algorithms. However, many real-world problems often possess complex characteristics, such as nonlinearity, discontinuity, multimodality, and non-differentiability, which make it difficult to formulate an accurate mathematical model or acquire prior knowledge by traditional mathematical techniques [20]. Therefore, researchers are now exploring approaches to solve optimization problems based on limited or no prior knowledge. This means developing algorithms that can adapt and learn from the problem itself, rather than relying heavily on explicit knowledge provided by experts or domain-specific information. For this, swarm-based algorithms have emerged as promising approaches to solve optimization problems in various fields [26]. Their ability to explore complex search spaces and find optimal or near-optimal solutions, even in the absence of prior knowledge, makes them valuable tools for tackling real-world challenges.

Swarm intelligence (SI) is a phenomenon in which simple agents, such as ants, bees and birds, work together to accomplish complex tasks that would be arduous or inaccessible for a single individual. This collective behavior involves decentralized decision-making, self-organization, adaptive responses to environmental dynamics, and the emergence of new properties that exceed the capabilities of individual agents. SI algorithms capture these principles and apply them to address different challenges in optimization, control, classification, clustering, routing, and prediction in various fields including engineering, robotics, biology, economics, social sciences and humanities [34]. These algorithms allow efficient exploration of large search spaces, convergence towards optimal or near-optimal solutions and the simultaneous management of several objectives or constraints. Additionally, they demonstrate scalability, robustness, fault tolerance and adaptability to dynamic or uncertain environments, thanks to their decentralized and distributed nature [9]. However, SI algorithms face challenges such as premature convergence, scalability issues, parameter sensitivity, and the need for rigorous theoretical foundations.

The main contribution of this paper is to provide future researchers and scholars with a roadmap to better understand current swarm-based systems and identify opportunities for further progress and improvement by exploring the underlying mechanisms of intelligence collective in natural swarms.

The paper is structured into four main sections: Section 2 explains the concept of SI, including its principles, design approaches and key characteristics. Section 3 presents an overview of SI algorithms. It covers their designs, uses and applications in other fields. The section also discusses the limitations of current SI approaches and highlights some of the challenges and potential directions for future research. Finally, Section 4 concludes the paper.

2. Swarm intelligence

Swarm Intelligence (SI) is a concept that originated in the field of cellular robotic systems, but has since expanded to various other areas, including optimization, sensor networks, data mining, machine learning, image processing, computer vision, etc. It is considered a branch of collective intelligence (Fig. 1), which falls under the broader framework of computational intelligence, which itself is a subset of artificial intelligence (AI) [41].

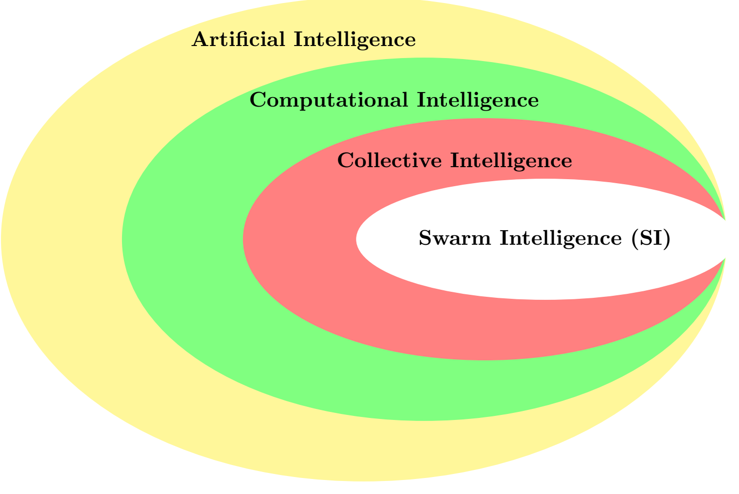

Figure 1. Where does “Swarm Intelligence (SI)” belong in the areas of Artificial Intelligence (AI)?

SI refers to the collective behavior exhibited by a group of relatively simple agents, often called ”particles” or ”individuals”, which are usually autonomous and have capabilities limited cognitive. They may have simple rules or behaviors that govern their actions, but they do not possess a global understanding or awareness of the system as a whole, but local interactions and communication between them generate collective intelligence.

2.1. Principles

When talking about SI, five fundamental principles are often considered guidelines for designing and understanding swarm behavior [32]. These principles are proximity, quality, diversity of responses, stability and adaptability (Fig. 2).

Proximity refers to the concept that agents in a swarm should have the ability to perceive and interact with their nearby neighbors. Local interactions allow them to share information, influence each other’s behavior and coordinate their actions. Proximity is crucial for effective communication and coordination within the swarm.

Quality refers to the performance or ability of individual agents based on specific criteria or objectives. Quality is measured based on the specific problem or task at hand. This may involve criteria such as efficiency, accuracy, timeliness, resource utilization or any other relevant indicator. Agents in a swarm must have mechanisms to measure their own performance, based on their local information and perception of the environment, and make decisions accordingly. Agents can contribute to the overall performance of the swarm by considering the quality of their own actions.

Diversity of responses refers to the variety of behaviors or strategies among agents in a swarm. If all agents follow the same behavior or strategy, the swarm may become vulnerable to obstacles or suboptimal solutions. By fostering diversity, the swarm can explore different possibilities and increase the chances of finding better solutions or adapting to changing environments.

Stability refers to the ability of a swarm to maintain its cohesion and functionality over time. This implies that the swarm must be resilient to disturbance, noise or disruption. A stable swarm can withstand external or internal variations and continue to exhibit consistent and efficient behavior. Stability ensures the robustness and reliability of the swarm.

Adaptability refers to the ability of a swarm to adapt and respond to changes in the environment or task requirements. An adaptive swarm can change its behavior, strategies, or structure to adapt to changing circumstances. Adaptability allows the swarm to handle dynamic and unpredictable situations and improves its overall performance and resilience.

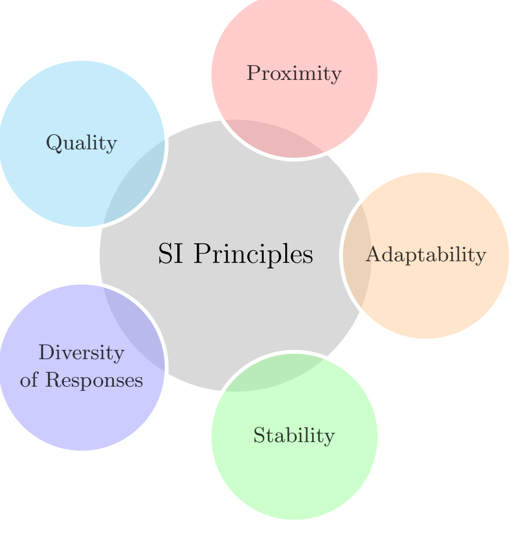

Figure 2. What properties must be present for a system to be considered a swarm-based system?

2.2. Design and features

The design and features of swarm systems encompass a range of aspects that define the fundamental characteristics and behavior of systems composed of multiple agents, called swarms. These aspects can be classified into those concerning individual agents and those concerning the entire swarm. Regarding individual agents, aspects include simplicity, limited memory, sensing capabilities, limited computing resources, and limited communication capabilities. As for the whole swarm, the aspects involve emergent behavior, scalability, decentralization, adaptability, communication protocols, fault tolerance, task allocation mechanisms, cooperation and collaboration, as well as robustness in the face of agent heterogeneity.

2.2.1. Agent aspects

Agent aspects in swarm systems refer to the unique features that distinguish individuals within the swarm.

First, the simplicity of the individual agent. The principle of swarm systems emphasizes that individual agents should be kept simple compared to a single sophisticated system serving the same goal. Agent simplicity helps reduce the complexity of each agent's behavior and control, making the swarm easier to manage and scale [18].

Second, limited memory. Basic agents in swarm systems are assumed to have limited memory. This means that the size of memory available to each agent remains constant regardless of the size of the problem or the number of agents in the system. Limited memory is a deliberate constraint placed on agents to simplify their design and control while promoting scalability and efficiency [10].

Third, sensing capabilities. The sensing capabilities of agents in swarm systems determine their perception of the environment and influence their decision-making process. These capabilities are often designed to be limited, consistent with the principle of simplicity of swarm systems. Agents generally have a local perception of the environment and base their decisions on the information available within their detection radius. This limitation allows for scalability and efficient coordination within the swarm, as agents only need to process and take into account local information rather than the entire global state of the environment [39].

Fourth, limited computing resources. The computing resources available to the agents are designed to be sufficient to execute the required algorithms and take into account their limited memory capacity [8].

Fifth, limited communication. Communication between agents in a swarm system is generally limited. A distinction is made between implicit and explicit communication, with implicit communication occurring as a side effect of other actions or via the environment. Explicit communication is a deliberate act aimed at conveying information to other agents. Explicit communication can take various forms, such as short-range point-to-point communication, global broadcasting, or the use of distributed shared memory [25].

2.2.2. Swarm aspects

Swarm aspects in swarm systems encompass properties related to the collective behavior and functionality of the entire swarm. First, collective intelligence. Swarm systems leverage the intelligence, knowledge, skills, and perspectives of all individuals in the swarm to solve complex tasks and make robust decisions that would be difficult for a single individual [15]. Second, scalability. Swarm systems provide the dynamism and flexibility to increase or decrease the number of individuals depending on the task at hand. This allows for efficient resource allocation and adaptability to different environments and mission requirements [40]. Third, redundancy and robustness. The existence of multiple individuals in swarm systems introduces positive redundancy, which improves reliability and robustness. Even if an individual fails or encounters malfunctions, the swarm can continue to function and accomplish its mission. This ensures the success of any mission, especially in harsh and dangerous environments [40]. Fourth, decentralization. Swarm systems operate on a decentralized principle, in which individuals interact locally with their neighbors without any centralized control or explicit communication. This decentralized approach reduces communication costs and improves system resilience. It also allows for rapid response to changes in the environment without the need for a hierarchical chain of command [15]. Fifth, emergent behaviors. Swarm systems can exhibit emergent behavior, where complex global behaviors emerge from simple local interactions between individuals. This emergent behavior allows the swarm to accomplish tasks such as exploration, foraging, or gathering without the need for explicit coordination. It contributes to the self-organizing and adaptive characteristics observed in natural swarms, leading to efficient and autonomous functioning [16]. Sixth, parallelism and task decomposition. Swarm systems benefit from parallelism. This is done by breaking down the overall task into subtasks. Each individual in a swarm performs one simultaneously, which helps speed up task execution. This approach allows for effective management of large-scale and time-sensitive tasks, especially tasks that would be impractical or impossible to accomplish alone [8]. Seventh, heterogeneity. Swarm systems can incorporate heterogeneous individuals with different physical properties and capabilities. These different individuals can complement each other, leveraging their unique abilities to optimize performance and achieve various goals [38].

3. SI algorithms

3.1. Overview

SI algorithms were designed taking inspiration from collective behaviors observed in diverse biological groups and swarms, ranging from simple organisms to complex social animals. Some examples include bees in composing and constructing hives, termites creating complex tunnel systems, ants finding paths when searching for food, birds flying in lines when searching for food [15]. These examples highlight how collective behaviors emerge in different biological swarms, allowing them to solve complex problems or achieve collective goals more effectively than individual members could. By studying and imitating these behaviors, SI algorithms aim to capture and replicate the intelligence and problem-solving abilities observed in natural swarms.

It is important to indicate that the Boids algorithm, developed by Craig Reynolds in the 1980s [37], is considered one of the first fundamental techniques in the field of SI, despite its origins in computer simulation rather than in direct application to the real world. The algorithm simulates the flocking behavior of birds (or "boids") by defining three simple local rules that each boid follows: separation, which refers to the avoidance of crowding, alignment, which indicates adaptation to the average velocity of neighbors, and cohesion, which means getting closer to the average position of neighbors. Through the collective application of these rules, the Boids simulation was able to generate remarkably realistic flocking models, demonstrating the principle of emergence, according to which complex global behaviors can arise from simple local interactions. While the Boids algorithm was initially designed to model flocks of birds, its underlying principles have since been applied to a wide range of other swarming entities, such as schools of fish, colonies of ants, etc., making it the cornerstone of SI research and a key example of how computer simulations can provide valuable insights into the fundamental mechanisms that determine collective behavior in natural and artificial systems.

SI algorithms are attracting great attention and popularity in the research community and beyond. This is due to four main reasons, including flexibility and versatility, efficiency in solving nonlinear design problems, mimicking nature, and paradigm-shifting potential [14]. SI algorithms have a high degree of flexibility and can be adapted to solve a wide range of optimization problems, including continuous, discrete, constrained, and multi-objective problems. This flexibility appears in the efficient exploration of the search space and their application in diverse fields, from engineering design and planning to finance and machine learning. SI algorithms have demonstrated exceptional performance in solving problems with highly nonlinear and complex objective functions and constraints. Indeed, the inherent parallelism and decentralized nature of these algorithms allow them to efficiently navigate complex search spaces, making them well suited to problems with multiple local optima and difficult landscapes.

SI algorithms can capture the elegance and efficiency of natural problem-solving strategies, which have evolved over millions of years. This biomimetic approach to optimization has captured the imagination of researchers and practitioners because it represents a more intuitive and biologically inspired way to solve complex problems. The development of truly intelligent algorithms based on the principles of SI, capable of adapting, learning and making decisions autonomously, represents an important potential paradigm shift in the field of optimization and computational intelligence. Such algorithms could lead to breakthroughs in areas such as real-time decision-making, adaptive control and self-organizing systems, with far-reaching implications across various industries and applications.

SI algorithms are a diverse and numerous set of algorithms that share a common framework. Although the details and specific steps may vary depending on the algorithm used, there is a general overview that describes the typical phases involved in many SI algorithms [11].

1. **Initialize the population**: The algorithm begins by creating an initial set of agents or particles that form the swarm. These agents represent potential solutions to the problem studied.
2. **Set the stopping condition**: The algorithm specifies the ending criteria, which determine when the algorithm should stop the iteration and return the final result. This may be a maximum number of iterations, reaching a certain level of solution quality, or other criteria.
3. **Evaluate the fitness function**: Each agent in the population is evaluated by a fitness function, which quantifies how well each agent solves the problem. The fitness function measures the objective or quality of each solution.
4. **Exploration**: The swarm focuses on exploring the search space. Agents perform random or diverse movements to explore different regions and discover potential solutions. This exploration phase helps the swarm avoid getting stuck in local optima and discover new areas of the search space.
5. **Exploitation**: The swarm focuses on exploiting promising regions identified during exploration. Agents adjust their behavior to intensify research in these areas, refining solutions and converging on better solutions.
6. **Update and move agents**: The positions or states of agents are updated based on certain rules or heuristics. These rules take into account both exploration and exploitation factors. Agents communicate and interact with each other, exchanging information to guide their movements and decision-making. This interaction helps the swarm collectively navigate the search space and improve the overall solution.
7. **Returns the best overall solution**: At the end of the execution of the algorithm, the swarm returns the best overall solution found. This solution represents the best solution discovered by the swarm over the iterations. It is the result of the collective intelligence and cooperation of the swarm agents.

Although this framework provides a general overview, the specific implementation and variations of each phase may differ across different SI algorithms. The effectiveness of a particular algorithm depends on the problem at hand and the specific strategies used in each phase.

There are many SI algorithms studied in the literature (Fig. 3). The diversity of organisms in nature has created a diversity of algorithms that mimic the interactions and cooperation observed in various species. Some of the most popular SI algorithms include Ant Colony Optimization (ACO) [19], Particle Swarm Optimization (PSO) [7], and Artificial Bee Colonies (ABC) [1]. However, there are many other SI algorithms that have been developed based on different intelligent behaviors exhibited by animals such as pigeons, fireflies, bees, birds, bats, etc. [20, 26] (see Tab. 1).

Figure 3. What are some examples of various SI algorithms?

Table 1
How do some SI algorithms behave?

Algorithm	Nature's behavior	Simulation	Algorithm process
Particle Swarm Optimization (PSO) [7]	Bird flock behavior, where individual birds adjust their movement based on their own experience and that of their neighbors	Particles represent potential solutions in a multidimensional search space. Particles adjust their velocities based on personal best and global best positions	1. Initialize particles with random positions and velocities. 2. Evaluate the fitness of each particle. 3. Update the personal best position for each particle. 4. Update the global best position based on the personal best positions of all particles. 5. Adjust the velocities and positions of particles based on personal and global best positions. 6. Repeat steps 2–5 until convergence or a specified number of iterations. 7. Extract the best particle position as the final result

Table 1 cont.

Algorithm	Nature's behavior	Simulation	Algorithm process
Ant Colony Optimization (ACO) [19]	Ant foraging behavior, where ants make pheromone trails to communicate and find the shortest paths to food sources	Virtual ants deposit pheromone trails on the graph nodes to represent the quality of the paths. Ants probabilistically choose paths based on pheromone levels and heuristic information	1. Initialize the pheromone trails at the edges of the graph. 2. Generate ant solutions by probabilistically selecting nodes. 3. Update pheromone trails based on ant solutions. 4. Repeat steps 2 and 3 until convergence or a specified number of iterations. 5. Extract the best ant solution as the final result
Artificial Bee Colony (ABC) [1]	Foraging behavior of bees, where bees explore food sources and communicate the information to other bees in the hive	Bees represent potential solutions and explore the search space using local search and global search phases	1. Initialize the bees with random positions. 2. Employ onlooker bees to choose food sources based on quality. 3. Update the best food source based on the quality of the chosen food sources. 4. Employ employed bees to explore the search space around chosen food sources. 5. Employ scout bees to randomly explore new food sources. 6. Repeat steps 2 through 5 until convergence or a specified number of iterations. 7. Extract the best food source as the end result
Bacterial Foraging Optimization (BFO) [22]	Bacterial foraging behavior, where bacteria move to nutrient-rich regions using chemotaxis and communicate through signaling	Bacteria are represented as solutions and chemotactic steps are simulated to adapt their positions	1. Initialize a population of bacteria with random positions. 2. Evaluate the fitness of each bacterium. 3. Simulate the chemotactic steps to adapt the positions of bacteria. 4. Perform reproduction and elimination steps based on the fitness of bacteria. 5. Repeat steps 2–4 until convergence or a specified number of iterations. 6. Extract the best bacterium position as the final result
Firefly Algorithm (FA) [41]	Flashing behavior of fireflies, where fireflies attract each other based on the brightness of their flashes	Fireflies represent potential solutions and adjust their brightness and positions to attract other fireflies	1. Initialize fireflies with random positions and brightness. 2. Evaluate the fitness of each firefly. 3. Update the brightness and positions of fireflies based on attraction and movement rules. 4. Repeat steps 2–3 until convergence or a specified number of iterations. 5. Extract the firefly with the highest brightness as the final result

Table 1 cont.

Algorithm	Nature's behavior	Simulation	Algorithm process
Cuckoo Search (CS) [41]	Brood parasitism behavior of cuckoo birds, where cuckoos lay eggs in the nests of other bird species and the host birds accept or reject the eggs	Cuckoos represent potential solutions and eggs represent solutions in the search space. Cuckoos lay eggs (new solutions) in nests (existing solutions) and host birds (evaluation function) determine acceptance or rejection of eggs	1. Initialize a population of cuckoos with random solutions. 2. Evaluate the fitness of each cuckoo. 3. Generate new solutions (eggs) by modifying cuckoo solutions. 4. Replace eggs in nests based on the host bird's acceptance rule. 5. Perform local random walk to explore new solutions. 6. Repeat steps 2–5 until convergence or a specified number of iterations. 7. Extract the best cuckoo solution as the final result
Pigeon-Inspired Optimization (PIO) [26]	Pigeon homing behavior, where pigeons navigate and return home using landmarks and orientation cues	Pigeons represent potential solutions and use landmarks to update their positions in the search space	1. Initialize a population of pigeons with random positions. 2. Evaluate the fitness of each pigeon. 3. Update the positions of pigeons based on landmarks and orientation cues. 4. Perform local search to refine the positions. 5. Repeat steps 2–4 until convergence or a specified number of iterations. 6. Extract the best pigeon position as the final result
Wolf Pack Algorithm (WPA) [26]	Social hierarchy and hunting behavior of wolf packs, where wolves collaborate to hunt and maintain a territory	Wolves represent potential solutions and interact based on social hierarchy and hunting strategies	1. Initialize a population of wolves with random positions. 2. Evaluate the fitness of each wolf. 3. Update the positions of wolves based on social hierarchy and hunting strategies. 4. Perform local search to improve the positions. 5. Repeat steps 2–4 until convergence or a specified number of iterations. 6. Extract the best wolf position as the final result
Artificial Fish-Swarm (AFS) [35]	Collective behavior of schools of fish, where fish adjust their positions based on their movements, availability of food, and avoidance of predators	Artificial fish represent potential solutions and swim in the search space using individual and collective rules of behavior	1. Initialize a population of artificial fish with random positions. 2. Evaluate the fitness of each fish. 3. Update the positions of fish based on individual movements, neighbor tracking, feeding behavior, and predator avoidance. 4. Perform local search to refine the positions. 5. Repeat steps 2–4 until convergence or a specified number of iterations. 6. Extract the best fish position as the final result

Table 1 cont.

Algorithm	Nature's behavior	Simulation	Algorithm process
Grey Wolf Optimizer (GWO) [31]	Social hierarchy and hunting behavior of gray wolf packs, where wolves collaborate to hunt and maintain a territory	Wolves represent potential solutions and interact based on social hierarchy and hunting strategies	1. Initialize a population of wolves with random positions. 2. Evaluate the fitness of each wolf. 3. Update the positions of wolves based on the alpha, beta, and delta wolf positions. 4. Perform local search to improve the positions. 5. Repeat steps 2–4 until convergence or a specified number of iterations. 6. Extract the best wolf position as the final result
Butterfly Optimization Algorithm (BOA) [30]	Foraging behavior of butterflies, where butterflies explore the search space using random flights and pheromone-mediated movements	Butterflies represent potential solutions and adjust their positions based on random flights and pheromone traces	1. Initialize a population of butterflies with random positions. 2. Evaluate the fitness of each butterfly. 3. Perform random flights to explore the search space. 4. Update the positions of butterflies based on pheromone trails. 5. Perform local search to refine the positions. 6. Repeat steps 2–5 until convergence or a specified number of iterations. 7. Extract the best butterfly position as the final result

3.2. Design and usage process

Designing and using swarm intelligence (SI) algorithms [41] involves following a general framework described as (see Fig. 4):

1. **Problem formulation**: Clearly define the optimization problem that the SI algorithm will solve. Specify the objective function, decision variables, constraints, and any other problem-specific considerations.
2. **Algorithm selection**: Review existing SI algorithms and select the one that best suits the problem at hand. Consider the strengths, weaknesses, and applicability of the algorithm to the problem domain.
3. **Initialization**: Set the initial state of the swarm, which includes the positions or solutions of individual agents. The initialization must cover the search space adequately to ensure diverse exploration of potential solutions.
4. **Agent behavior and interactions**: Determine the behavior of individual agents (e.g., particles, ants, or birds) within the swarm. This involves defining how agents make decisions, update their positions, communicate with each other, and adapt their behavior based on local and global information.
5. **Objective evaluation**: Develop a method to evaluate the quality of each agent's solution based on the objective function. This evaluation feature guides the search process by providing feedback on the suitability or optimality of each solution.

6. **Swarm dynamics**: Define the rules that govern interactions and dynamics within the swarm. This involves determining how agents share information, exchange knowledge, and adjust their behavior based on the collective intelligence of the swarm.
7. **Parameter tuning**: Identify algorithm parameters (e.g., swarm size, convergence criteria, communication range) that affect algorithm performance. Perform sensitivity analysis or use optimization techniques to find appropriate values for these parameters based on the characteristics of the problem.
8. **Convergence analysis**: Analyze the convergence properties of the algorithm to understand its convergence speed and the quality of the solutions obtained. Use mathematical tools and techniques to prove convergence to globally optimal or near-optimal solutions under certain assumptions.
9. **Experimental validation**: Implement the algorithm and conduct experiments to validate its performance. Compare the algorithm's results with other state-of-the-art approaches or known optimal solutions to evaluate its effectiveness and efficiency.
10. **Improvement strategies**: Explore strategies to improve the performance of the algorithm. This may include integrating problem-specific knowledge, introducing adaptive mechanisms, exploring hybridization with other optimization techniques, or taking into account dynamic variations of the problem.
11. **Iterative improvement**: Based on the experimental results and information obtained, refine and iterate the algorithm design. Consider changes to agent behavior, swarm dynamics, parameters, or other aspects to improve the algorithm's convergence properties, efficiency, and solution quality.
12. **Benchmarking**: Compare the improved algorithm with existing approaches and benchmark it against relevant performance metrics. This analysis helps understand the strengths and weaknesses of the algorithm and provides insight into its applicability to different problem domains.

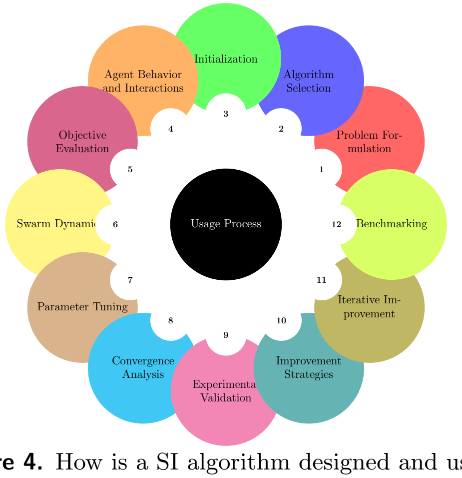

Figure 4. How is a SI algorithm designed and used?

3.3. Applications

Swarm Intelligence (SI) algorithms have gained popularity due to their diverse applications in various fields. Some of the key applications of SI algorithms include machine learning, engineering, assignment, routing, networking, bioinformatics, etc. [1, 15, 19, 30, 31, 35].

Figure 5. How powerful are swarm intelligence algorithms?

3.3.1. Synergies

Table 2
How can SI algorithms create synergies with other scientific fields?

Field	Synergy examples
Machine Learning (ML)	In [7], the authors explore the challenges of high dimensionality and low sample size in cancer classification using microarray datasets. The study introduces an adaptive feature selection method that combines three filters (Chi-square, Information Gain and ReliefF) at first to reduce dimensionality and accelerate the training process. In the second stage, SI algorithms, particularly particle swarm intelligence (PSO), are used to select optimal features to improve classification performance. Finally, Ensemble learning techniques with different classifiers are applied in parallel with PSO to evaluate the model performance. The proposed method aims to improve the accuracy and efficiency of classification in cancer diagnosis and treatment

Table 2 cont.

Field	Synergy examples
	In [43], the authors investigate the use of two swarm intelligence algorithms, the Sparrow Search Algorithm (SSA) and the Whale Optimization Algorithm (WOA), to optimize the performance of an Extreme model Gradient Boosting (XGBoost) to predict the penetration rate (PR) of Tunnel Boring Machines (TBM) in complex geological conditions. Predicting TBM penetration rate is a critical task in metro construction projects because it helps engineers plan the construction schedule, estimate costs, and manage risks more effectively. The SSA algorithm mimics the foraging behavior of sparrows, while the WOA algorithm draws inspiration from the hunting techniques of humpback whales. In this study, researchers integrated these swarm intelligence algorithms with the XGBoost model, a powerful machine learning technique for regression and classification tasks. The objective of the SSA and WOA algorithms was to explore and identify optimal hyperparameter combinations for the XGBoost model, leading to more reliable and accurate predictions of TBM performance
Data science and Big Data	In [28], the researchers propose a big data text clustering algorithm based on swarm intelligence to address the limitations of current clustering algorithms and avoid the impact of disturbances on anomalous big data text clustering. They build a differential privacy model based on the characteristics of swarm intelligence, such as distribution and self-organization. The goal of integrating swarm intelligence principles is to provide a flexible data conversion platform to handle the incomplete information structure of big data, build a differential privacy protection model using KD tree partitioning to protect user location data while retaining data utility, and process location information via dimensionality reduction and clustering. In [12], the authors propose a Big Data-based model to address the limitations of traditional logistics systems and achieve more efficient management of e-commerce logistics warehouses, particularly through the optimization of goods allocation and warehouse routes. They used a Multi-Objective Particle Swarm Optimization (MOPSO) approach to optimize warehouse allocation and solve the problem of slow allocation of goods in warehouses. The study introduces a dynamic mutation probability formula to overcome the problem of convergence of local optima in traditional models. Ant Colony Optimization algorithms based on Genetic Algorithms (GA-ACO) were also used to develop a logistics warehouse path optimization model, thereby obtaining optimized logistics warehouse paths. All this to address the limitations of traditional logistics systems to meet the growing demands of e-commerce logistics, including efficient merchandise planning, convenient accessibility, equipment integration and optimization of merchandise management
Natural Language Processing (NLP)	In [21], the authors address the problem of topic modeling, which is a fundamental text analysis technique used to extract underlying topic structures or "topics" from a collection of text documents. Traditionally, topic modeling has been approached using single-objective techniques such as Latent Dirichlet Allocation (LDA), which optimizes a single criterion such as perplexity to discover topics. However, the authors argue that topic modeling is inherently a multi-objective problem, as there are several desirable properties that one would like to optimize simultaneously, such as coherence, coverage, and perplexity. For that, they propose a multi-objective optimization approach to topic modeling using a swarm intelligence algorithm called the Multi-Objective Artificial Bee Colony (MOABC) algorithm

Table 2 cont.

Field	Synergy examples
	MOABC is designed to simultaneously optimize the three objectives of consistency, coverage and perplexity, rather than aggregating them into a single objective function. This allows the user to choose the most appropriate compromise between the different objectives
Robotics	In [42], the authors present a novel decentralized and asynchronous robotic search algorithm based on Particle Swarm Optimization (PSO) to solve mazes and find targets in complex unknown environments. The proposed algorithm uses robots as particles in a PSO algorithm and equips them with toolkits to change course and avoid obstacles, as well as to memorize and reuse their best personal experiences to avoid dead ends. The algorithm is completely decentralized, requiring minimal communication between robots and no central synchronization. Robots move and update asynchronously. The fitness function is simply the inverse Euclidean distance to the target, requiring minimal knowledge of the environment on the part of the robots. The performance of the proposed algorithm remains constant even if the complexity of the search environment increases, unlike some other methods
Networking and distributed systems	In [44], the authors discuss the optimization of vehicle routing problem (VRP) in logistics network routing using an improved Pigeon-Inspired Optimization (PIO) algorithm. VRP is a fundamental problem in logistics and transportation that aims to plan the route of vehicles to meet customer demands while minimizing total costs. The study presents the new PIO algorithm, which exploits the strengths of the quantum evolutionary algorithm to improve global exploration and a Gaussian variation operator to improve local exploitation and prevent premature convergence. The improved PIO algorithm is designed in such a way that each individual contains information about customer points and routes. The objective is to optimize VRP in logistics networks to satisfy the ‱5R principle" (right quality, right quantity, right price, right time, right route) and minimize total logistics and distribution costs
Embedded systems	In [33], the authors discuss the application of multi-objective optimization techniques based on swarm intelligence to optimize the operation of industrial cooling towers to improve energy efficiency. Cooling towers are a crucial part of refrigeration systems in power plants and large buildings, used to dissipate heat and cool process water. Growing concerns regarding environmental sustainability and efficient use of energy and water resources justify the need to optimize the operation of cooling towers. The study proposes to use multi-objective optimization algorithms based on swarm intelligence, such as Multi-Objective Particle Swarm Optimization (MOPSO), to find the optimal operational set points for cooling towers. The objectives are to maximize the efficiency of the cooling tower while minimizing the overall power consumption of the refrigeration system, subject to operational constraints
Quantum computing	In [17], the authors present a novel approach that integrates quantum machine learning and deep self-learning techniques to solve the problem of emergency transportation management during the COVID-19 crisis. The authors first develop a quantum version of the OPTICS clustering algorithm, called Quantum OPTICS (QOPTICS), which aims to improve the computational efficiency of the classical OPTICS algorithm

Table 2 cont.

Field	Synergy examples
	They then propose in-depth self-learning approaches for two swarm intelligence algorithms, Artificial Orca Algorithm (AOA) and Elephant Herd Optimization (EHO). These deep self-learning variants, called DSLAOA and DSLEHO, use dynamic mutation operators to improve the efficiency of the original swarm algorithms. To leverage both efficiency and effectiveness, the authors further hybridize deep self-learning swarm algorithms with the QOPTICS algorithm, allowing the swarm algorithms to operate on a single identified cluster by QOPTICS. This hybrid approach is then applied to the real-world problem of optimizing emergency vehicle dispatch and patient transportation during the COVID-19 pandemic, with the goal of minimizing response times and ensuring adequate coverage in different geographic regions
Bioinformatics and computational biology	In [36], the authors propose a swarm intelligence-based hierarchical clustering approach to identify non-coding RNAs (ncRNAs) using a covariance search model. Covariance models (CMs) have been effective in identifying potential members of existing ncRNA families, but they have some drawbacks, such as being computationally expensive and limited to family-specific searches. Previous work used Hierarchical Agglomerative Clustering (HAC) to combine overlapping CMs into a single combined CM (CCM), but this eliminates structural information and dilutes sequence features as more families are added. The authors propose a novel approach that uses Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) to select the best base pairs among multiple CMs to construct CCMs, with the aim of improving the performance of discernibility
Cloud computing	In [24], the authors present a novel method to improve task scheduling in healthcare services based on cloud computing in the IoT environment. This method is a new hybrid optimization algorithm called HPSOSSA which combines the strengths of Particle Swarm Optimization (PSO) and the Salp Swarm Algorithm (SSA). Task scheduling is a crucial challenge in cloud-based HCS because it impacts the timely satisfaction of user requests, service delivery costs, and service quality. For this, HPSOSSA algorithm is designed to solve this complex problem and then reduce factors such as lifespan, waiting time and resource utilization and finally improve healthcare service delivery
Distributed Constraint Optimization Problem (DCOP) [2–6]	In [23], the authors propose a new algorithm to solve the distributed constraint optimization problem (DCOP) using Particle Swarm Optimization (PSO). DCOP is a fundamental problem in multi-agent systems where agents cooperatively solve a constraint optimization problem. However, DCOP is NP-hard, making it difficult to solve large-scale problems in real time. The PSO process forms groups of agents, which are then used to solve the DCOP in a distributed and cooperative manner. The key innovation is to exploit swarm intelligence to discover appropriate optimality criteria, rather than imposing them a priori. This aims to provide a more flexible and adaptive DCOP algorithm capable of handling complex and large-scale multi-agent optimization problems. In [13], the authors present a novel Ant-based algorithm called ACO-DCOP for solving the distributed constraint optimization problem (DCOP). DCOP constitutes an important framework in multi-agent systems where agents must coordinate their decisions to optimize an overall objective function

Table 2 cont.

Field	Synergy examples
	The study presents new mechanisms within the ACO framework that are tailored to the DCOP context, including a way to calculate heuristic factors that capture local benefits and a method to calculate pheromone deltas that take into account the cost structure of DCOP. The study provides theoretical analysis showing that ACO-DCOP is an anytime algorithm, meaning it can return solutions of increasing quality over time

3.4. Limitations

It is important to consider a set of limitations and address them appropriately when applying SI in real-world applications. First, sensitivity to parameter settings and initial conditions. SI algorithms can be very sensitive to initial configuration and parameter settings. Slight changes in these factors can lead to suboptimal solutions or convergence failure, especially in large-scale systems [29]. Second, vulnerability to disruption. Swarms may be resilient to individual failures, but may be vulnerable to systemic disruptions such as environmental changes, resource depletion, or external attacks. These disruptions can destabilize the swarm, leading to disintegration, divergence, or oscillations [8]. Third, the balance between exploration and exploitation. Swarms must find a balance between exploring the space for new solutions and exploiting the best solutions found. There is a risk of getting stuck in optimal or suboptimal local regions if the swarm lacks diversity or adaptability [27]. Fourth, scalability. Although swarms can scale to large numbers of agents, the computational and communication costs can become prohibitive in large-scale systems. Efficient algorithms for coordination, decision-making and resource allocation are required, which can be difficult to design and optimize [8]. Fifth, social and ethical considerations. When using SI algorithms in social media or human-centric domains, there is a risk that swarm behavior will inadvertently reinforce the existing beliefs of individuals within the system. For example, if the Swarm algorithm favors or prioritizes certain types of content or interactions based on user preferences or prior engagement, this may lead to biased presentation of information. This bias can further reinforce the existing beliefs, opinions, or ideologies of individuals within the system. There is also swarm polarization, which refers to the division of individuals or groups into distinct factions with extreme or divergent beliefs. When SI algorithms reinforce existing beliefs and limit exposure to diverse perspectives, it can contribute to the creation of communities where individuals become more entrenched in their own views and less open to alternative ideas. This polarization can lead to increased animosity, reduced empathy, and a breakdown in constructive dialogue [8].

3.5. Challenges

Although SI algorithms have grown in popularity, they also expose several key challenges that researchers are actively addressing in this field [30, 31, 41]. First, despite the impressive performance of SI algorithms in practical applications, there is often a mismatch between the theoretical understanding of these algorithms and their observed effectiveness. Researchers are working to develop more rigorous theoretical frameworks to analyze the behavior and convergence properties of SI algorithms. Advances in mathematical modeling, complexity analysis and the development of new analytical tools are crucial to improve the theoretical foundations of SI algorithms and explain their empirical success.

Second, SI algorithms each have a diverse range of techniques and approaches, each with their own terminology and classification schemes. This lack of standardization can hinder communication and collaboration between researchers, as it can be difficult to compare and understand different algorithms and their underlying principles. Efforts are underway to standardize key terminology used in this area. This can facilitate better understanding, knowledge sharing and progress in the research community.

Third, the performance of SI algorithms often strongly depends on the choice of their parameters, such as the number of particles, the inertial weight or the rate of disappearance of pheromones. Finding optimal parameter configurations can be a difficult optimization problem in itself, because the performance of algorithms can be sensitive to the values of these parameters. Researchers are exploring various techniques, such as adaptive parameter control, machine learning-based methods, and self-adaptive mechanisms, to automate the parameter tuning process and improve the robustness of SI algorithms.

Fourth, although SI algorithms have demonstrated success in solving smaller-scale optimization problems, their applicability to large-scale, complex real-world problems remains a significant challenge. Scaling these algorithms to address large-scale problems with high dimensionality, many constraints, and massive search spaces requires new approaches and strategies. Researchers are investigating ways to improve the scalability of SI algorithms, such as hybridization with other optimization techniques, use of parallel and distributed computing, and development of problem-specific modifications.

Fifth, with the proliferation of various SI algorithms, researchers are often faced with the challenge of selecting the most appropriate algorithm for a given optimization problem. The choice of algorithm depends on factors such as problem characteristics, desired performance criteria, available computational resources, and specific application requirements. The development of guidelines, benchmarking frameworks, and decision support tools can help researchers and practitioners navigate the landscape of SI algorithms and make informed choices for their specific optimization tasks.

3.6. Future directions of research

Swarm intelligence (SI) algorithms have emerged as an important class of optimization techniques, inspired by the collective behavior of natural swarms. However, the field of SI algorithms faces several critical future research directions that researchers are actively investigating. First, conducting rigorous theoretical analysis is crucial to improve the understanding of the underlying principles, convergence properties and optimization capabilities of SI algorithms. This includes studying convergence behavior, analyzing complexity, and establishing formal mathematical frameworks for analyzing and comparing different algorithms. Second, researchers are exploring effective hybridization strategies that combine multiple SI algorithms or integrate them with other optimization methods, with the aim of achieving a better balance between exploration and exploitation, avoiding premature convergence, and improving the quality of solutions [24]. Solving complex optimization problems, such as high-dimensional, multimodal, and dynamic problems, is another key area of interest, as researchers adapt SI algorithms to address these challenges more effectively. Additionally, the development of a unified optimization framework for SI algorithms is a promising direction, as it could provide a systematic understanding of their performance differences, inspire the development of new algorithms, and facilitate the selection of appropriate algorithms for specific optimization problems. Finally, adopting a multidisciplinary approach by integrating knowledge from biology, psychology, computer science and engineering can lead to the creation of more generic and versatile optimization frameworks that draw on from various disciplines. By addressing these future research problems, the field of swarm intelligence algorithms can continue to evolve, providing more powerful and efficient optimization solutions for a wide range of real-world applications.

4. Conclusion

The paper provided an overview of Swarm Intelligence (SI) and its basic algorithms, highlighting their fundamental principles, design features and broad applications. It identified several common patterns or frameworks that contribute to the effective problem-solving capabilities of swarms, such as decentralized decision-making, use of simple local rules, redundancy and fault tolerance, and exploitation of indirect communication through the environment. These principles have inspired the development of various SI algorithms and distributed problem-solving approaches in areas such as robotics, computing, and logistics, providing new ways to address complex challenges in a decentralized and adaptive manner.

The paper also highlighted the need to address remaining challenges and explore promising future research directions in swarm intelligence. Improving the theoretical foundations, developing robust hybrid approaches integrating swarm intelligence with other optimization techniques, and improving the performance, simplicity, and versatility of swarm-based optimizers are identified as key areas for future research. By addressing these directions, future work on swarm intelligence can pave the way for even broader real-world applications and continued advancements in this dynamic and rapidly evolving field, unlocking the full potential of swarm-based solutions to solve complex optimization problems.

References

[1] Abu-Mouti F.S., El-Hawary M.E.: Overview of Artificial Bee Colony (ABC) algorithm and its applications. In: *2012 IEEE International Systems Conference SysCon 2012*, pp. 1–6, IEEE, 2012. doi: 10.1109/syscon.2012.6189539.

[2] Adrdor R., Ezzahir R., Koutti L.: Consistance d'arc souple appliquée aux problèmes DCOP. In: Z. Bouraoui, S. Doutre (eds.), *Plate-Forme Intelligence Artificielle: Actes des 14es Journées d'Intelligence Artificielle Fondamentale*, pp. 63–72, 2020. https://hal.science/hal-02951644v1/file/Actes_JIAF_PFIA2020.pdf#page=63.

[3] Adrdor R., Koutti L.: Enhancing AFB_BJ$^+$-AC* algorithm. In: *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, pp. 1–7, IEEE, 2019. doi: 10.1109/ICCSRE.2019.8807711.

[4] Adrdor R., Koutti L.: Asynchronous Forward-Bounding algorithm with Directional Arc Consistency. In: *ASPOCP 2021: Workshop on Answer Set Programming and Other Computing Paradigms 2021 co-located with ICLP 2021 Porto, Portugal, September 21*, 2021. https://ceur-ws.org/Vol-2970/aspocppaper7.pdf.

[5] Adrdor R., Koutti L.: Enforcing Full Arc Consistency in Asynchronous Forward Bounding Algorithm, *Journal of Communications Software and Systems*, vol. 18(1), pp. 9–16, 2022. doi: 10.24138/jcomss-2021-0083.

[6] Adrdor R., Koutti L.: Improvement of Arc Consistency in Asynchronous Forward Bounding Algorithm. In: *AJCAI 2021: The 34th Australasian Joint Conference on Artificial Intelligence, 2–4 February 2022, Sydney*, pp. 582–591, Springer, 2022. doi: 10.54985/peeref.2304p2837660.

[7] Alrefai N., Ibrahim O.: Optimized feature selection method using particle swarm intelligence with ensemble learning for cancer classification based on microarray datasets, *Neural Computing and Applications*, vol. 34(16), pp. 13513–13528, 2022. doi: 10.1007/s00521-022-07147-y.

[8] Altshuler Y.: Recent Developments in the Theory and Applicability of Swarm Search, *Entropy*, vol. 25(5), 710, 2023. doi: 10.3390/e25050710.

[9] Altshuler Y., Pentland A., Bruckstein A.M.: *Swarms and network intelligence in search*, Springer, 2018. doi: 10.1007/978-3-319-63604-7.

[10] Boley D., Gini M., Zhang Y.: How do robot swarms behave? What graphs can tell us. In: *ARMS workshop at 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), May 29, 2023, London, United Kingdom*, 2023.

[11] Brezočnik L., Fister Jr I., Podgorelec V.: Swarm intelligence algorithms for feature selection: A review, *Applied Sciences*, vol. 8(9), 1521, 2018. doi: 10.3390/app8091521.

[12] Chen Z., Liu J., Wang Y.: Big Data Swarm Intelligence Optimization Algorithm Application in the Intelligent Management of an E-Commerce Logistics Warehouse, *Journal of Cases on Information Technology*, vol. 26(1), pp. 1–19, 2024.

[13] Chen Z., Wu T., Deng Y., Zhang C.: An ant-based algorithm to solve distributed constraint optimization problems. In: *Thirty-Second AAAI Conference on Artificial Intelligence*, vol. 32, 2018. doi: 10.1609/aaai.v32i1.11580.

[14] Cobo A., Llorente I., Luna L.: Swarm intelligence in optimal management of aquaculture farms. In: *Handbook of Operations Research in Agriculture and the Agri-Food Industry*, pp. 221–239, Springer, 2015. doi: 10.1007/978-1-4939-2483-7_10.

[15] 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*, vol. 21(6), 2062, 2021. doi: 10.3390/s21062062.

[16] Diggelen van F., Luo J., Karagüzel T.A., Cambier N., Ferrante E., Eiben A.E.: Environment induced emergence of collective behavior in evolving swarms with limited sensing. In: *GECCO '22: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 31–39, 2022. doi: 10.1145/3512290.3528735.

[17] Drias H., Drias Y., Houacine N.A., Bendimerad L.S., Zouache D., Khennak I.: Quantum OPTICS and deep self-learning on swarm intelligence algorithms for Covid-19 emergency transportation, *Soft Computing*, vol. 27(18), pp. 13181–13200, 2023.

[18] Efremov M.A., Kholod I.I.: Swarm robotics foraging approaches. In: *2020 IEEE conference of Russian young researchers in electrical and electronic engineering (EIConRus)*, pp. 299–304, IEEE, 2020. doi: 10.1109/eiconrus49466.2020.9039340.

[19] Fidanova S.: Ant colony optimization. In: *Ant Colony Optimization and Applications*, Studies in Computational Intelligence, vol. 947, pp. 3–8, Springer, Cham, 2021. doi: 10.1007/978-3-030-67380-2_2.

[20] Figueiredo E., Macedo M., Siqueira H.V., Santana Jr C.J., Gokhale A., Bastos-Filho C.J.A.: Swarm intelligence for clustering : A systematic review with new perspectives on data mining, *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 313–329, 2019. doi: 10.1016/j.engappai.2019.04.007.

[21] González-Santos C., Vega-Rodríguez M.A., Pérez C.J.: Addressing topic modeling with a multi-objective optimization approach based on swarm intelligence, *Knowledge-Based Systems*, vol. 225, 107113, 2021. doi: 10.1016/j.knosys.2021.107113.

[22] Guo C., Tang H., Niu B., Lee C.B.P.: A survey of bacterial foraging optimization, *Neurocomputing*, vol. 452, pp. 728–746, 2021. doi: 10.1016/j.neucom.2020.06.142.

[23] Hasegawa K., Noto M.: Swarm intelligence algorithm for optimality discovery in distributed constraint optimization. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3611–3616, IEEE, 2014. doi: 10.1109/smc.2014.6974490.

[24] Hassan K.M., Abdo A., Yakoub A.: Enhancement of health care services based on cloud computing in IoT environment using hybrid swarm intelligence, *IEEE Access*, vol. 10, pp. 105877–105886, 2022. doi: 10.1109/access.2022.3211512.

[25] Hou K., Yang Y., Yang X., Lai J.: Cooperative control and communication of intelligent swarms: A survey, *Control Theory and Technology*, vol. 18(2), pp. 114–134, 2020. doi: 10.1007/s11768-020-9195-1.

[26] Hu J., Wu H., Zhong B., Xiao R.: Swarm intelligence-based optimisation algorithms: An overview and future research issues, *International Journal of Automation and Control*, vol. 14(5-6), pp. 656–693, 2020. doi: 10.1504/ijaac.2020.10030986.

[27] Kwa H.L., Babineau V., Philippot J., Bouffanais R.: Adapting the Exploration–Exploitation Balance in Heterogeneous Swarms: Tracking Evasive Targets, *Artificial Life*, vol. 29(1), 2022. doi: 10.1162/artl_a_00390.

[28] Li X., Shu Z.: Research on Big Data Text Clustering Algorithm Based on Swarm Intelligence, *Wireless Communications and Mobile Computing*, vol. 2022(1), 7551035, 2022. doi: 10.1155/2022/7551035.

[29] Lin J.H., Yeh M.C.: A Swarm Intelligence Approach to Parameters Identification of Chaotic Systems. In: *2006 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3509–3514, IEEE, 2006. doi: 10.1109/icsmc.2006.384663.

[30] Makhadmeh S.N., Al-Betar M.A., Abasi A.K., Awadallah M.A., Doush I.A., Alyasseri Z.A.A., Alomari O.A.: Recent advances in butterfly optimization algorithm, its versions and applications, *Archives of Computational Methods in Engineering*, vol. 30(2), pp. 1399–1420, 2023. doi: 10.1007/s11831-022-09843-3.

[31] Makhadmeh S.N., Al-Betar M.A., Doush I.A., Awadallah M.A., Kassaymeh S., Mirjalili S., Zitar R.A.: Recent advances in Grey Wolf Optimizer, its versions and applications, *IEEE Access*, vol. 12, pp. 22991–23028, 2023. doi: 10.1109/ACCESS.2023.3304889.

[32] Millonas M.M.: Swarms, phase transitions, and collective intelligence, *arXiv preprint adap-org/9306002*, 1993.

[33] Nedjah N., Mourelle L.d.M., Lizarazu M.S.D.: Swarm Intelligence-Based Multi-Objective Optimization Applied to Industrial Cooling Towers for Energy Efficiency, *Sustainability*, vol. 14(19), 11881, 2022. doi: 10.3390/su141911881.

[34] Neme A., Hernández S.: Algorithms inspired in social phenomena. In: *Nature-inspired algorithms for optimisation*, pp. 369–387, Springer, 2009. doi: 10.1007/978-3-642-00267-0_13.

[35] Pourpanah F., Wang R., Lim C.P., Wang X.Z., Yazdani D.: A review of artificial fish swarm algorithms: Recent advances and applications, *Artificial Intelligence Review*, vol. 56(3), pp. 1867–1903, 2023. doi: 10.1007/s10462-022-10214-4.

[36] Pratiwi L., Choo Y.H., Muda A.K., Pratama S.F.: Swarm Intelligence-based Hierarchical Clustering for Identification of ncRNA using Covariance Search Model., *International Journal of Advanced Computer Science and Applications*, vol. 13(11), pp. 822–831, 2022. doi: 10.14569/ijacsa.2022.0131195.

[37] Reynolds C.W.: Flocks, herds and schools: A distributed behavioral model. In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 25–34, 1987. doi: 10.1145/280811.281008.

[38] Sasaki H.: Emulating heterogeneity of individuals and visualizing its influence on ant swarm migration, *Applied Artificial Intelligence*, vol. 36(1), 2138120, 2022. doi: 10.1080/08839514.2022.2138120.

[39] Shahzad M.M., Saeed Z., Akhtar A., Munawar H., Yousaf M.H., Baloach N.K., Hussain F.: A review of swarm robotics in a nutshell, *Drones*, vol. 7(4), 269, 2023. doi: 10.3390/drones7040269.

[40] Wilson J., Chance G., Winter P., Lee S., Milner E., Abeywickrama D., Windsor S., *et al.*: Trustworthy Swarms. In: *TAS '23: Proceedings of the First International Symposium on Trustworthy Autonomous Systems*, 2023. doi: 10.1145/3597512.3599705.

[41] Yang X.S., Karamanoglu M.: Swarm intelligence and bio-inspired computation: an overview, *Swarm Intelligence and Bio-Inspired Computation*, pp. 3–23, 2013. doi: 10.1016/B978-0-12-405163-8.00001-6.

[42] Youssefi K.A.R., Rouhani M.: Swarm intelligence based robotic search in unknown maze-like environments, *Expert Systems with Applications*, vol. 178, 114907, 2021. doi: 10.1016/j.eswa.2021.114907.

[43] Yu Z., Li C., Zhou J.: Tunnel Boring Machine Performance Prediction Using Supervised Learning Method and Swarm Intelligence Algorithm, *Mathematics*, vol. 11(20), 4237, 2023. doi: 10.3390/math11204237.

[44] Zhang X., Wei Y., Hashim Z.: Improvement of Swarm Intelligence Algorithm and Its Application in Logistics Network Routing, *Journal of Network Intelligence*, vol. 8(4), pp. 1077–1094, 2023. https://bit.kuas.edu.tw/~jni/2023/vol8/s4/02.JNI-S-2023-04-014.pdf.

Affiliations

Rachid Adrdor
Ibn Zohr University, Faculty of Sciences, Department of Computer Science, Agadir, Morocco, rachid.adrdor@edu.uiz.ac.ma

Received: 22.05.2024
Revised: 1.06.2024
Accepted: 5.06.2024