



Article

Inverse Firefly-Based Search Algorithms for Multi-Target Search Problem

Ouarda Zedadra ^{1,*} , Antonio Guerrieri ^{2,*} , Hamid Seridi ¹ , Aymen Benzaid ³ and Giancarlo Fortino ^{2,4}

¹ LabSTIC Laboratory, Department of Computer Science, 8 Mai 1945 University, P.O. Box 401, Guelma 24000, Algeria; seridi.hamid@univ-guelma.dz

² CNR, National Research Council of Italy, Institute for High Performance Computing and Networking (ICAR), Via P. Bucci 8/9C, 87036 Rende, Italy; giancarlo.fortino@unical.it

³ Department of Computer Science, 8 Mai 1945 University, P.O. Box 401, Guelma 24000, Algeria; aymen14madrid14@gmail.com

⁴ Department of Informatics, Modeling, Electronics, and Systems, University of Calabria, 87036 Rende, Italy

* Correspondence: zedadra.ouarda@univ-guelma.dz (O.Z.); antonio.guerrieri@icar.cnr.it (A.G.)

Abstract: Efficiently searching for multiple targets in complex environments with limited perception and computational capabilities is challenging for multiple robots, which can coordinate their actions indirectly through their environment. In this context, swarm intelligence has been a source of inspiration for addressing multi-target search problems in the literature. So far, several algorithms have been proposed for solving such a problem, and in this study, we propose two novel multi-target search algorithms inspired by the Firefly algorithm. Unlike the conventional Firefly algorithm, where light is an attractor, light represents a negative effect in our proposed algorithms. Upon discovering targets, robots emit light to repel other robots from that region. This repulsive behavior is intended to achieve several objectives: (1) partitioning the search space among different robots, (2) expanding the search region by avoiding areas already explored, and (3) preventing congestion among robots. The proposed algorithms, named Global Lawnmower Firefly Algorithm (GLFA) and Random Bounce Firefly Algorithm (RBFA), integrate inverse light-based behavior with two random walks: random bounce and global lawnmower. These algorithms were implemented and evaluated using the ArGOS simulator, demonstrating promising performance compared to existing approaches.

Keywords: swarm intelligence; swarm robotics; multi-targets search problem; firefly algorithm; random bounce; global lawnmower



Citation: Zedadra, O.; Guerrieri, A.; Seridi, H.; Benzaid, A.; Fortino, G. Inverse Firefly-Based Search Algorithms for Multi-Target Search Problem. *Big Data Cogn. Comput.* **2024**, *8*, 18. <https://doi.org/10.3390/bdcc8020018>

Academic Editor: Miguel-Angel Sicilia

Received: 1 November 2023

Revised: 12 February 2024

Accepted: 14 February 2024

Published: 19 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Swarm robotics represents an innovative approach to coordinating large numbers of robots, drawing inspiration from the collective behaviors observed in social insects [1]. It represents the application of Swarm Intelligence (SI) within Multi-Robot Systems (MRSs). This paradigm emphasizes the significance of physical embodiment and realistic interactions among robots and their environment. Swarm robotics is characterized by the emergence of synchronized behaviors at the system level, even when individual agents may be relatively limited, centralized coordination is absent, and interactions are kept simple [2].

Multi-robot exploration involves deploying a group of robots with the task of searching for multiple targets dispersed in an unknown environment. In such environments, employing a random walk strategy presents a viable choice as it offers the potential to discover more targets. However, this approach can be time-consuming, as robots may repeatedly cover already explored areas. Moreover, there is a risk that robots may not reach the most distant areas, thereby reducing the likelihood of finding additional targets. A widespread dispersion of robots facilitates the exploration of more areas, thereby increasing

the chances of locating more targets. Another alternative is distributing the search areas among robots, enabling rapid exploration of environments. However, achieving this in swarms of robots without direct communication can be challenging.

The practical implementation of multi-target search and foraging algorithms poses challenges in swarm robotics, as there remains a disparity between proposed algorithms and their actual deployment. One contributing factor to this disparity could be the communication methods employed among robots. For instance, while stigmergic communication is often proposed, its real-world application is often limited by cost considerations. In contrast, light-based communication has emerged as a promising alternative. This method entails wireless data transmission through light, where Light-Emitting Diodes (LEDs) emit light pulses modulated with data. Light-based communication facilitates the transmission of various data types, including position information, sensor data, and commands. Compared to radio-based communication, light-based communication offers several advantages in the context of swarm robotics search tasks:

1. It avoids bandwidth problems compared to radio-frequency communication.
2. It offers a more reliable and interference-free communication channel compared to other electronic devices and radio signals.
3. It is an energy-efficient communication mechanism designed specifically for robots with limited power. This will expand the life of the robots and thus provide longer mission duration.
4. The signals can be directed toward specific robots. This will help in improving communication efficiency and reduce interference.
5. Light-based communication can scale to large swarm sizes without performance degradation.

In this paper, we propose two lightweight multi-target search algorithms named *Random Bounce Firefly Algorithm (RBFA)* and *Global Lawnmower Firefly Algorithm (GLFA)*. These algorithms integrate inverse light-based behavior with two random walks: random bounce and global lawnmower. Light-based communication is used here to realize the following three main objectives:

1. Ensuring robots disperse widely throughout their environment to maximize object discovery;
2. Implicitly partitioning the search space among robots through indirect communication;
3. Ensuring effective local exploration.

The proposed algorithms are implemented in the multi-robots ArGOS simulator. The obtained results are compared with two other related algorithms (LFA [3] and the original Firefly algorithm (FF) [4]). The comparisons are promising and prove the superiority of the proposed algorithms in some simulation scenarios.

The remainder of the paper is structured as follows: we present related works and background algorithms in Section 2. In Section 3, we introduce the finite state machine and provide the pseudo-code for the proposed algorithms. Following this, we present and discuss the experimental results in Section 4. Section 5 highlights real-world applications and limitations, while Section 6 offers insights into implementing the proposed algorithms with real robots. Finally, the paper concludes in Section 7.

2. Related Works

In multi-robot exploration, it is crucial for the robots to cover various search areas to explore the environment and create a useful map effectively. This requirement applies to diverse fields, including industrial applications, like automated lawnmowers or vacuum cleaners, military operations, such as mine clearance, and humanitarian efforts like search and rescue operations [5]. This section summarizes relevant works related to multi-target search and compares them with our proposal.

The paper by Palmieri et al. [6] presents two multi-robot algorithms: “Firefly-based Team Strategy for Robot Recruitment (FTS-RR)” and “Ant-based Team Strategy for Robot

Recruitment (ATS-RR)". These algorithms aim to address the challenge of locating and disarming distributed mines. FTS-RR combines Ant Colony Optimization (ACO) and Firefly (FF) algorithms, while ATS-RR relies solely on ACO. In this strategy, robots disperse into cells and independently explore using ACO, leaving pheromones on visited cells. Upon detecting a mine, robots are mobilized for disarmament. The detecting robot acts as a firefly, attracting others based on intensity values influenced by distance. Additionally, a cooperating robot becomes a coordinator, attracting robots with different pheromones to guide them to mine locations. Simulations show that FTS-RR outperforms in terms of mission completion time, cell visits, and overall robot distribution, leading to more efficient disarmament times.

Dimidov et al. [7] conducted a comprehensive study on random walk models for Kilobots searching for a static target under various environmental conditions. The research focused on a swarm of robots searching for a single static target, with the N robots varying from 10 to 100. Communication among robots was limited to a radius of 10 cm, and a multi-agent simulation abstracted the kilobots' physical details. Experiments in a confined 90 cm square arena revealed that the Coherent Random Walk (CRW) with high persistence was most effective in bounded spaces, while the Levy Walk (LW) performed poorly due to collisions. LW emerged as the optimal strategy in unlimited space, with some movement correlation providing an advantage. Additionally, the distance from the target to the central location significantly influenced system performance, with LW demonstrating scalability in information dissemination.

The authors in [8] propose a hybrid approach that integrates the FTS-RR and ATS-RR algorithms [6] to tackle the challenge of mine detection and disarmament using a robotic team. Compared to other algorithms like PSO and ATS-RR, FTS-RR has exhibited superior performance. Experiments conducted by the authors using FTS-RR in a Java-based grid environment involved varying parameters, such as the number of targets, robots, and the size of the environment. The findings emphasize the importance of parameter values, as task complexity increases with larger environments and more mines. Specifically, achieving a balance between the attractiveness of the fireflies and introducing random movements is crucial. This balance enhances robot distribution, mitigates repetition in specific areas, and reduces the time required to complete all tasks.

In Katada et al. [9], the authors tackle the challenge of multi-target detection in enclosed environments using two exploration algorithms: the conventional Random Walk (RW) and Lévy Flight (LF). Their proposed subsumption architecture comprises three layers: transmission, obstacle avoidance, and target exploration. The authors conducted experiments with three prototypes: random walk with step sizes of 1 (RN2) and 6 (RN6), and Lévy Flight (LF6). LF6 involves two phases: forward movement and rotation, with step sizes determined by a Lévy probability distribution. The target, an infrared emitter ball, is placed in the lower left corner, while a wireless station is positioned in the upper right corner. The robots commence from a fixed initial position near the station and are oriented randomly. The target is located 80 m away, and robots must maintain connectivity with the station. Experiments involved varying the number of robots (10, 15, 20) and concluded when the station receives a message or after 1800 s without a message. The exploration strategy changed among RN2, RN6, and LF6, with ten experiments for each configuration. The success rate was notably high, with LF6 demonstrating the best performance among the explored strategies.

Palmieri et al. [10] conducted a comparative analysis of hybridized algorithms for mine detection and disarmament in a distributed environment. They compared the hybridization of Ant Colony Optimization (ACO) and Firefly algorithm (FF) with Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC). The three hybrid proposals (ACO-FF, ACO-PSO, and ACO-ABC) were evaluated using a Java-based simulator with robots having limited energy stores. ACO-FF uses a combination of ACO and FF for search and collaboration in mine disarmament, as previously explained in [6]. A coordination mechanism was introduced in ACO-ABC, where robots employed ABCs to determine movement decisions when receiving

multiple requests. *ACO-PSO* uses the *PSO* algorithm for mine disarmament upon detection. Simulations revealed higher energy consumption with the *PSO* approach, especially with small swarm sizes and high task complexity. *FF* and *ABC* methods were compatible with less complex tasks, but differences became apparent with more targets and smaller robot swarms. In complex tasks, the coordination mechanism became more intricate, and the *FF*-based strategy outperformed in terms of performance.

The authors in [11] focus on the application of the algorithms proposed in [6] in simulated environments and compare them to the *PSO* algorithm. The authors assessed the solution's quality by analyzing a specific performance metric—the relative error indicating the number of accesses in the cells, providing a measure of the efficiency with which the robots are distributed in the area. The simulations show that the relative error is low by applying the *ACO* and the *FF* algorithms together.

In Suarez et al. [12], the authors present a new exploration strategy inspired by bats, specifically microbats that use sonar for navigation in the dark. This algorithm involves a prime number of individuals (P) representing bats, each assigned a location x_i and a velocity v_i in the search space. The algorithm initializes these variables randomly and calculates each bat's pulse frequency and volume. The swarm evolves through generations, each representing a temporal state, until reaching the maximum number of generations G_{max} . New frequency, position, and speed are calculated for each generation g and each bat. The current best solution is determined by evaluating the objective function for all bats, and local solutions are selected based on specific criteria. The algorithm then intensifies the search through a local random walk, perturbing the chosen solution from the best current solutions. Experimental results demonstrate effectiveness, particularly in scenarios involving challenges like complex configurations (e.g., U and V shapes), traffic jams, or narrow passages. The authors highlight the adaptability of the bat-inspired strategy in navigating challenging environments.

In Santos et al. [13], the authors introduce a novel exploration strategy utilizing a continuous multi-modal utility function and metaheuristic optimization. They frame the exploration problem as an optimization task, generating data for a *Firefly-based algorithm (FF)* tasked with searching solution space cells to construct a map. The environment is represented as a *two-dimensional* objective function, with minimum energy points as potential robot targets. A metaheuristic, based on *FF*, seeks to minimize the objective function, directing the robot towards optimal targets. The exploration task includes searching boundaries for maximum information gain, devising collision-minimizing paths based on map characteristics, and avoiding previously explored areas to minimize redundancy. These directives are weighted in a behavior, defining the objective function as the weighted sum of their component functions for position selection. Considering computational efficiency, *FF* is utilized for objective function minimization. The strategy is implemented on the *V-REP* robotic platform under the *ROS (Robot Operating System)*. Tests conducted in 10 m environments utilize conditional, linear, and propositional loss metrics, with propositional loss effectively guiding time and trajectory length exploration.

Abuomar and Al-Aubidy propose [14] an enhanced swarm robotics-based search and rescue algorithm, a refinement of the Binary Dragonfly Optimization Algorithm (*BDA*). This improved version, the Robotic Binary Dragonfly Algorithm (*RBDA*), integrates two additional behaviors: obstacle avoidance and communication. The original *BDA* includes five fundamental behaviors: obstacle avoidance, speed matching, approaching a center, attraction to an area of interest, and fleeing from negative influences. The *RBDA* algorithm is tailored to adapt to environments with obstacles and communication constraints through adjustments to the step vector equation. Evaluation is performed using Benchmark functions (*sphere*, *Rosenbrock*, *De Jong*, *Grie Wank*, *Rastrifin*, *Ackliu*) and compared against the original *BDA*. Results reveal that *RBDA* minimizes error and converges more effectively to optimal solutions. Simulations are executed using the *SIMROBOT* virtual Toolbox in Matlab, with a simulated environment spanning 300×300 m and featuring randomly distributed obstacles. The number of robots ranges from 5 to 15, with a victim robot randomly

placed. RBDA displays enhanced performance in achieving optimal solutions, with faster convergence times observed as the number of robots increases.

The authors of [15] address the challenge of a multi-robot target search. They improve the *ATS-RR strategy* (explained in [6]) by integrating a nature-inspired distributed wireless communication protocol. This protocol facilitates on-site recruitment of the required robot, minimizing traffic. When a robot detects a target, it sends message announcements among the robots. Simulation results demonstrate reduced time compared to the *Inverse Ant System-Based Surveillance System (IAS-SS)*. Moreover, increasing the number of targets prolongs the convergence time.

In Khaluf et al. [16], the *Collective Lévy Walk (CLW) algorithm* is introduced for robot swarms, utilizing local communication to create a collective Lévy walk pattern. The algorithm prioritizes longer steps in robot trajectories by exchanging information between robots. *CLW* is a deterministic automaton where robots start exploration in a running state, moving at a constant linear speed while determining the duration of their next step (*TL*) from a Lévy distribution. Upon completing the *TL interval*, robots enter a spinning state, rotating at a *constant angular velocity (TU)* sampled from a uniform distribution. If an obstacle is detected via proximity sensors, a walking robot switches to collision avoidance behavior, rotating based on the obstacle's distance, and then returns to the walking state after avoidance, sampling a new *TL interval* to continue the Lévy walk. The algorithm's crucial aspect is utilizing communication among robots to generate a collective Lévy walk, where each robot broadcasts its sampled *TL* to local neighbors within its communication range and sector of vision. Steps are categorized as short or long based on a predefined threshold (*F*). Simulations conducted in a $20 \times 20 m^2$ arena using the ARGoS simulator demonstrate the algorithm's efficacy. Results, averaged over 30 executions lasting 5000 time steps each, show *CLW's* ability to generate collective trajectories even for larger swarm sizes. Simulation parameters include an exponent (α) set to 1, robot communication range of 1.35 m, linear speed of 5 m/s, and a step threshold (*F*) set to 1.53 m (with 0.17 m as the simulated robot's diameter).

In Huang et al. [17], the authors focus on implementing an exploration scenario involving a swarm of heterogeneous robots capable of detecting various types of targets simulating different radiation levels. The study investigates the influence of experimental parameters on swarm efficiency using the *Mona* simulation platform. The exploration algorithm comprises three steps: (1) random walk, where the robot continuously explores the environment at a constant speed using its sensory system; (2) obstacle detection, where the robot adjusts motor speeds based on sensors to change direction and continues the random walk upon detecting obstacles; (3) target detection, where the robot halts to investigate detected targets, records their relative positions, and resumes random walking. Simulations involve preventive radiation, petroleum, or oil drums in *V-REP*, each with distinct radiation levels requiring separate detection. Three versions of the *Mona* robot are designed, each capable of detecting a specific type with different colors. Performance evaluation is based on the total number of target detections and the time to locate all targets. Two simulation scenarios explore the impact of increasing robot numbers and changes in environmental configurations. Results indicate a slight increase in the total number of detected targets with more robots. Statistical analysis using *ANOVA* reveals that environmental complexity does not significantly affect performance.

In Pang et al. [18], the authors introduce an advanced random exploration strategy where robots adjust their step sizes dynamically based on the estimated density of robots in specific areas. This adaptive approach aims to evenly distribute robots in the environment, thereby enhancing exploration efficiency. The study includes both computer simulations using the *Webots* simulator and experiments with real robots (*e-puck*). Results from simulations demonstrate the superiority of the proposed method over existing strategies such as *Brownian Walk (BW)* and *Lévy Flight (LF)* in terms of coverage ratio. Real experiments also show the effectiveness of the proposed method compared to *BW* and *LF*, measured by the number of found objects.

In [19], the authors introduce a novel multi-robot exploration strategy named *Distributed Particle Swarm Optimization (DPSO)* designed for search and rescue operations. Their focus lies in enabling a swarm of robots to effectively navigate the search space, avoid collisions and obstacles, and locate victims. The *DPSO* algorithm incorporates two primary functions: the artificial potential function, which attracts forces to unknown areas and victims, and the repulsion forces function, aiding in collision avoidance. The repulsion function comprises intra and inter-repulsion forces. The experiments were conducted by implementing a multi-agent model and environment using Python and *VRep*. Results from the experiments validate that the proposed *DPSO* algorithm assists robots in navigating away from local minima, enabling them to discover alternative paths through disaster scenarios to reach optimal solutions.

In [20], a search strategy named *Velocity-inspired Robotic Fruit Fly Algorithm (VRFA)* is introduced for expediting victim search and enabling real-time assessment and management of search and rescue operations. This strategy combines the *Fruit Fly Optimization algorithm (FOA)* for searching and tracking multiple targets and the *Particle Swarm Optimization (PSO) algorithm* for updating the position and velocity of fruit flies. Each independent robot engages in four activities: local recording and data processing, inter-robot data sharing, development of a movement plan, and creation of timelines for executing movement. The algorithm is evaluated alongside other techniques such as *PSO*, *FOA*, and *BSO*, using two cooperation strategies (centralized and decentralized) for static and dynamic targets. The study investigates the impact of the number of targets, environment complexity, and the number of robots on system performance. Results indicate that decentralized cooperation enhances performance, with *VRFA* demonstrating the best average performance in reducing search and rescue time.

In [21], a novel neural network algorithm using the *Artificial Bee Colony (ABC)* method is introduced to tackle the challenges of full-coverage path planning in search and rescue operations. This algorithm aims to comprehensively cover the area of interest within a constrained time frame and autonomously navigate around obstacles in unfamiliar terrains. The neural network takes inputs about obstacles and coverage in five directions and produces the speed of the left and right wheels as output. By leveraging the advantages of the *ABC algorithm*, such as increased probability of finding the optimal solution and system robustness, the authors optimize the parameters of the neural network, enhancing the overall training efficiency and effectiveness. Performance evaluation of the algorithm is based on an evaluation function divided into three parts: coverage rate, path repetition rate, and failure rate. Experimental results show that integrating the *ABC* method with a neural network path planning algorithm effectively guides rescue robots to plan comprehensive coverage paths. Additionally, this approach exhibits high robustness across diverse environments.

Table 1 has been populated to highlight all the features of the reviewed work. According to such a table, several observations can be made:

Table 1. Qualitative comparison of the summarized related works.

		[18]	[19]	[13]	[14]	[9]	[20]	[6]	[11]	[8]	[10]	[21]	[15]	[17]	[12]	[7]	[22]	[16]	
Environment	Complexity	with obstacles		X	X	X		X			X	X	X	X				X	
		Free of obstacles	X				X		X	X	X						X	X	
Object	Distribution	Random	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	
		partially clustered																X	
		Completely clustered																	X
	Nature	Static		X	X	X	X		X	X	X		X	X	X				
		Dynamic						X											
Robot	Position	Central	X																
		Random	X	X	X		X												
		Predefined					X						X						
	Energy	limited	X				X		X			X	X	X	X				
		unlimited		X	X	X		X		X	X		X	X				X	X
	Sensors	limited	X	X	X	X	X	X	X			X	X	X		X	X	X	
	Memory	No							X	X		X		X	X	X	X		
Yes		X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	
Homogeneity	No													X					
	Direct		X		X (Wifi)	X (Wifi)	X					X	X (Wifi)			X	X		
Strategy	Communication	Indirect						X	X	X	X	X	X		X		X		
		Inspiration						Fruit Fly	ACO/FF	ACO/FF	ACO/FF	ACO/FF/PSO/ABC	ABC, ANN	ACO				X	
	Exploration	Random	X	X			X								X	X	X	X	
		Strategic	X	X	X	X			X	X	X	X	X	X				X	
		Inspiration	LF, BM	PSO	FF	Dragonfly		Fruit Fly	ACO	ACO	ACO	ACO	ABC, ANN	ACO				DDSA	
	Recruitment	Yes							X	X	X	X	X	X	X				
		No	X			X													
	Inspiration							ACO/FF	ACO/FF	FF	FF/PSO/ABC	FF	ACO	bat					

- The majority of the studies used an obstacle-free environment.
- Most of the studies employed static targets with random distribution.
- Only a few studies fixed the position of the robots, such as [18], which followed both central and random strategies; [9], where the robots had predefined positions; and [14], which followed the random strategy.
- Half of the studies considered unlimited robot energy, while the other half considered limited energy.
- The majority of the studies considered limited robot perception.
- All the studies used homogeneous robots, most of which used without memory.
- Most of the studies employed bio-inspired algorithms as exploration strategies and direct and indirect communication behaviors for the robots, with the majority also utilizing collaborative work.
- The experiments were tested either in the real world, such as [16] that used e-puck robots and [22], or in simulations on platforms like ARGoS, Player/Stage, Gazbo, etc., and other custom-developed platforms.
- The studies using the FF algorithm utilize its attractive light behavior. None of the studies based on the firefly algorithm considered light to repel the robots.

Previous works have primarily employed the Firefly (FF) algorithm to attract robots to regions of interest. To the best of our knowledge, our study marks the first and only instance where the FF algorithm is utilized to repel robots from each other within covered areas. Upon discovering a resource region, a robot emits light to deter other robots from approaching. This repulsion mechanism ensures several benefits: (1) Implicit distribution of robots across the search space, enhancing the likelihood of discovering additional regions; (2) prevention of congestion and collisions in search areas; (3) expedited exploration of local regions, facilitated by the algorithm’s utilization of spiral search patterns.

3. The Proposed Algorithms

In this work, we propose two swarm intelligence-based algorithms to resolve the multi-target search problem. We choose the Firefly algorithm (FF) [23] to fit our contributions in using light-based communication. Unlike the related works in the literature, the light here is used to repulse robots from regions under exploration. This repulsive behavior guarantees an implicit division of the search space between robots and ensures efficient exploration. In unknown environments, the random walk is important since it allows the robots to explore far regions. In the proposed algorithms, we combine the FF with Random Bounce (RB) random walk producing the *Random Bounce Firefly Algorithm (RBFA)* and the FF with Global Lawnmower (GL) random walk to produce the *Global Lawnmower Firefly Algorithm (GLFA)*. The two proposed algorithms, GLFA and RBFA, use the same finite state machine illustrated in Figure 1.

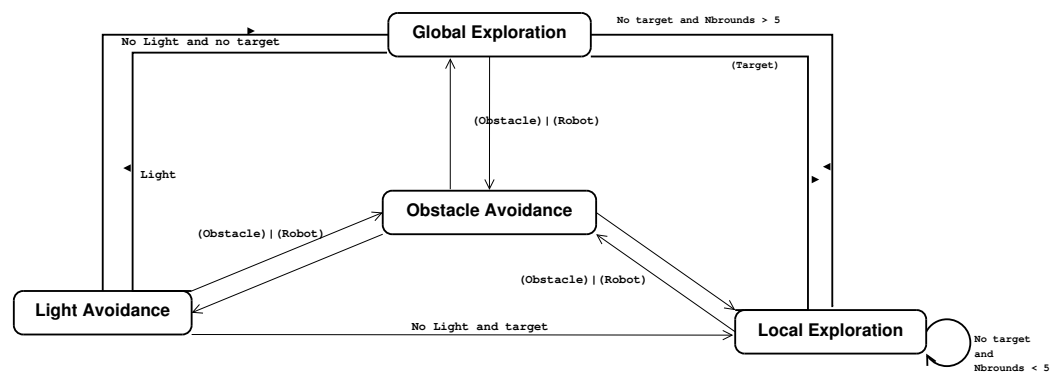


Figure 1. Finite state machine of Random Bounce Firefly Algorithm (RBFA) and Global Lawnmower Firefly Algorithm (GLFA) Robots.

In the proposed algorithms, robots start with the *Global Exploration state* (Algorithm 1) since no obstacle, target, or light is detected. In RBFA, robots use the *random bounce algorithm* and in GLFA, the *global lawn-mower algorithm*. When a target is found, the robot switches to the *local exploration state* (Algorithm 2) and: (1) destructs the found target and increases its number, (2) decreases the number of rounds, (3) increases the light intensity, and (4) continues its local search. After five rounds, if no target is found, the robot switches to the *global exploration state* using RB or GL Algorithms (Algorithm 1). When light is detected at a distance d (Equation (4)), the robot switches to the *light avoidance state* (Algorithm 3), where it changes its direction, and increases its velocity to move away from the emitted light source. If a target is found in its path, it switches to the *local exploration state* (Algorithm 2), or else it switches to the *global exploration state* (Algorithm 1). From any state, if an obstacle is encountered, the robot switches to the *obstacle avoidance state* (Algorithm 4). A detailed textual description and the pseudo-codes of the different states are given below.

Algorithm 1: Global exploration using RB or GL

```

1 while ( $\nexists$ obstacle,  $\nexists$ target,  $\nexists$ light) do
2   | Explore the environment using RB or GL random walks.
3 if ( $\exists$  Obstacle) then
4   | Goto Obstacle Avoidance() (Algorithm 4);
5 else
6   | if ( $\exists$  target) then
7     | Goto Local Exploration() (Algorithm 2);
8   | else
9     | if ( $\exists$  light) then
10    | | Goto Light Avoidance (Algorithm 3);

```

Algorithm 2: Local exploration algorithm

```

1 while ( $nb_{rounds} < 5$ ) do
2   | Explore the environment locally using the Square Spiral Search Algorithm.
3 if ( $\exists$  Target) then
4   | Destruct the found target;
5   | Increase the number of found targets;
6   | Increase the light intensity using Equation (2);
7   | disperses the light intensity using Equation (3);
8   | Decrease the number of rounds;
9 else
10  | if ( $\exists$  Obstacle) then
11    | Goto Obstacle Avoidance () (Algorithm 4);
12  | else
13    | Goto Global Exploration() (Algorithm 1);

```

Algorithm 3: Light avoidance algorithm

```

1 while distance  $d$  is reached do
2   | Change direction to 90 degrees;
3   | Double velocity using Equation (5);
4 if ( $\exists$  Obstacle) then
5   | Goto Obstacle avoidance () (algorithm 4);
6 else
7   | if ( $\exists$  target) then
8     | Goto Local Exploration () (Algorithm 2);
9   | else
10    | if ( $\nexists$  light) then
11      | Goto Global Exploration() (Algorithm 1);

```

Algorithm 4: Obstacle avoidance algorithm

```

1 Retrieve readings from all proximity sensors;
2 Calculate the value  $V$  of the maximum reading  $L_{max}$ ;
3 if ( $V > 0$ ) then
4   | calculate the angle  $A$  of  $L_{max}$ ;
5 if ( $A$  shows that the obstacle is in front of the robot) then
6   | Walk backwards;
7 else
8   | if ( $A$  shows that the obstacle is on the left side) then
9     | Move right and choose a random angle;
10  | else
11    | Move left and choose a random angle

```

1. **Global Exploration:** In this state, robots explore their space, searching for targets. They use Random Bounce (RB) or Global Lawnmower (GL) random walks. When a robot locates a target, it switches to the *local exploration state*. A brief description of RB and GL algorithms is given below:

- The random bounce algorithm is based on a random walk where each robot moves randomly in different directions according to Equation (1). However, when encountering an obstacle, a robot, or a predefined boundary, it bounces in a random direction to avoid the obstacle [24].

$$\Theta_{new} = \Theta + n \quad (1)$$

In Equation (1), Θ represents the robot's direction at the time of detection, n is a uniform random variable, and the distribution is determined by which side of the robot encountered the obstacle, thereby triggering the rotation.

If the robot detects something on the left side, it adheres to a distribution $n \sim \mathcal{U}(\frac{\pi}{4}, \frac{3\pi}{4})$, on the right side $n \sim \mathcal{U}(-\frac{3\pi}{4}, -\frac{\pi}{4})$, and at the center $n \sim \mathcal{U}(\frac{3\pi}{4}, \frac{5\pi}{4})$.

- The global lawnmower algorithm involves a systematic search method where robots move in parallel lines, covering the entire search area in a way similar to mowing a lawn. This method ensures complete region coverage but may be less efficient in complex environments [25]. The pattern encompasses two primary movements: (1) a straight path and (2) a semi-circular path. The 'Pitch' denotes the spacing between two consecutive straight trajectories. Corners and edges are often missed and overlooked due to the rotation. In fact, many types of searches suffer from this dependency.

2. **Local exploration:** When locating a target, the robot: (1) executes a local exploration using a square spiral search, (2) increases the light intensity using Equation (2), (3) disperses the light intensity according to the distance from the source using Equation (3).

$$I_{t+1} = I_t \times T \quad (2)$$

In Equation (2), I_t represents the intensity of light at time t , and T represents the value of the increase in intensity.

$$I_{(r)} = I_t \times K \quad (3)$$

In Equation (3), $I_{(r)}$ represents the intensity of light at a distance r , and K represents the value of the increase in distance.

3. **Obstacle avoidance:** When an obstacle is detected, the robot alters its direction to avoid it. It returns to the same state if possible; otherwise, it switches to the *global exploration state*.
4. **Light avoidance:** When the robot detects that it is within a distance d (Equation (4)) from the source of emitted light, it changes its direction of 90 degrees and moves away using a new velocity according to Equation (5) until no light is detected, then the robot changes to the *global exploration state*.

$$d = \sqrt{\frac{k}{I_{max}}} \quad (4)$$

In Equation (4), d is the distance from the light source; k represents the initial intensity; I_{max} is the intensity at which the robot should start avoiding light,

$$v_{i+1} = v_i \times z_i \quad (5)$$

In Equation (5), v_i is the speed of the robot at the current time, and z_i is the coefficient of inertia of the robot.

4. Experimental Analysis

The experimental simulation was conducted using the multi-physics robot simulator ARGoS [26], known for its efficiency in simulating large-scale swarms of diverse robots. Several computer simulations were executed to evaluate the performance of the proposed algorithms. The robots use their sensors to capture shapes, colors, distances, and environmental features. After that, they process the collected data to recognize and identify the objects using a computer vision algorithm with some predefined criteria. In our simulations, targets are represented by a green 3D cylinder with dimensions of $2.5 \text{ cm} \times 10 \text{ cm}$.

4.1. Simulation Scenarios

Table 2 displays the considered simulation scenarios. Each simulation was executed for a duration of 20 min . In all the simulations, the obstacle density was set to 10%. Given that we are using stochastic algorithms, the results presented in the following experiments are the averages of 10 simulations for each scenario. The four scenarios implemented were designed to investigate the impact of the following variations, respectively: (i) the number of robots, (ii) the size of the environment, (iii) the number of clusters, and (iv) the distribution of targets (clustered or uniform). For the scenarios run, the parameters are detailed in Table 2.

Table 2. The parameters used in the simulation scenarios.

Parameter	Value
Scenario 1: Variation in the number of robots	
Number of Robots	30, 40, 50, 60 robot
Target Distribution	clustered
Number of Clusters	12 cluster
Number of Targets	910 target
Environment Size	120 m × 120 m
Scenario 2: Variation in the environment size	
Environment Size	80 × 80, 100 × 100, 140 × 140, 200 × 200
Number of Robots	50 robot
Target Distribution	clustered
Number of Targets	910 target
Number of Clusters	12 cluster
Scenario 3: Variation in the number of clusters	
Number of Clusters	2, 4, 8, 16 cluster
Number of Robots	50 robot
Target Distribution	clustered
Number of Targets	1000 target
Environment Size	120 m × 120 m
Scenario 4: Variation in the distribution of targets	
Target Distribution	clustered, uniform
Number of Robots	50 robot
Number of Targets	600 target
Number of Clusters	12 cluster
Environment Size	120 m × 120 m

4.2. Results and Discussion

We conducted a comparative analysis of the proposed algorithms (GLFA, RBFA) with Lévy walk and Firefly Algorithm (LFA [3]) and the Firefly Algorithm (FF [4]). In the subsections below, we will present the results obtained in various scenarios and discuss and analyze the outcomes.

4.2.1. Results of Scenario 1

In this scenario, our focus lies in investigating how varying the number of robots influences the performance of the RBFA, GLFA, LFA, and FF algorithms. The number of robots ranges from 30 to 60, with the results summarized in Table 3 and depicted in Figure 2. As the number of robots increases, the percentage of found targets also rises. For instance, with 30 robots, RBFA located 77.38% of the targets, while LFA found 28.42%, GLFA discovered 65.97%, and FF located 36.02%. With each incremental increase in the number of robots, algorithm performance improves as robots cover more areas of the environment. With 60 robots, RBFA identified 87.71% of the targets, GLFA found 78.43%, whereas LFA and FF found 41.9% and 1.20% respectively. In conclusion, RBFA and GLFA consistently outperform LFA and FF algorithms. FF's reliance on random walks results in slower target discovery, while LFA's attractive behavior limits exploration opportunities. Conversely, the repulsive behavior in GLFA and RBFA fosters broader exploration, leading to higher target discovery rates.

Table 3. Percentage of found targets when increasing the number of robots—Scenario 1.

Num. of Robots	30	40	50	60
RBFA	77.38%	82.35%	84.50%	87.71%
GLFA	65.98%	67.67%	69.58%	78.44%
LFA	28.42%	32.92%	37.4%	41.9%
FF	0.28%	0.49%	0.80%	1.20%

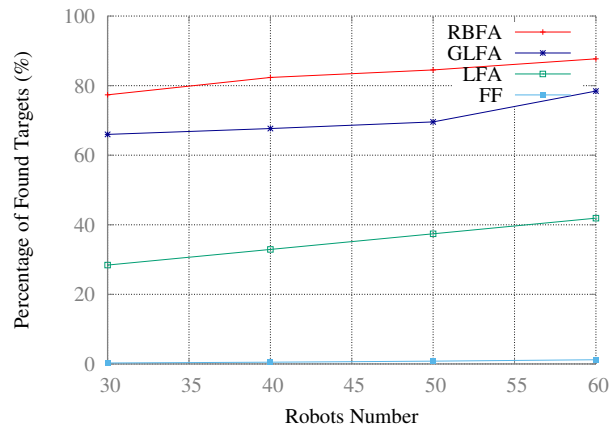


Figure 2. Percentage of found targets when increasing the number of robots—Scenario 1.

4.2.2. Results of Scenario 2

This scenario investigates how the environment’s size impacts the proposed algorithms’ efficacy. In each successive simulation, we progressively expanded the environment size from 80 m² to 200 m². The results are summarized in Table 4 and illustrated in Figure 3. Across all four algorithms, the percentage of discovered targets decreased as the environment size increased. The RBFA and LFA algorithms yielded notably superior results compared to the GLFA and FF algorithms. In RBFA, including repulsive behavior facilitated exploration into distant regions, albeit at the expense of spending more time on exploration rather than target exploitation. Conversely, in the LFA algorithm, the attractive behavior enhances the likelihood of target exploitation over exploration, particularly in larger environments.

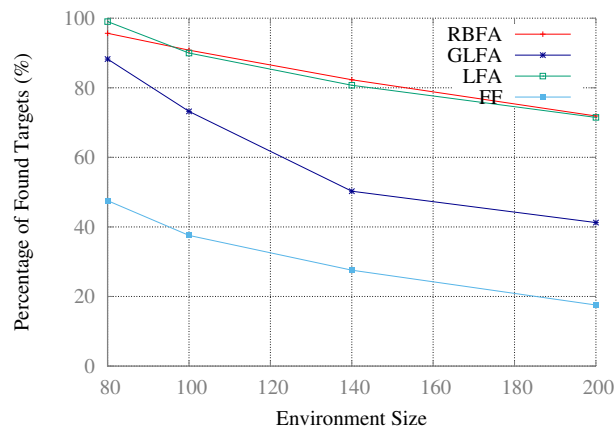


Figure 3. Percentage of found targets when increasing the size of the environment—Scenario 2.

Table 4. Percentage of found targets when increasing the size of the environment—Scenario 2.

Env. Size	80 × 80	100 × 100	140 × 140	200 × 200
RBFA	95.65%	90.81%	82.31%	71.88%
GLFA	88.24%	73.21%	50.26%	41.25%
LFA	99%	89.99%	80.73%	71.48%
FF	47.56%	37.56%	27.56%	17.56%

4.2.3. Results of Scenario 3

This scenario investigates the impact of varying the number of clusters on the performance of the four algorithms. Targets are grouped into clusters of different sizes to determine if distributing them across multiple clusters enhances algorithm performance. The number of clusters is adjusted from 2 to 16, doubling in each new simulation. The results, as depicted in Table 5 and Figure 4, show that the percentage of found targets increases with more clusters. RBFA and LFA yield closely comparable results. LFA performs better with 2 and 4 clusters, while RBFA outperforms with 8 and 16 clusters. This distinction arises because the LFA algorithm prioritizes exploitation upon target discovery, leading to superior results with smaller clusters (2 and 4) where exploitation opportunities are more abundant. Conversely, with larger clusters (8 and 16), the limited number of targets diminishes the efficacy of exploitation, favoring RBFA’s emphasis on exploration. In RBFA, encouraging exploration into other regions facilitates exploitation by individual robots.

Table 5. Percentage of found targets when increasing the number of clusters—Scenario 3.

Clusters Num.	2	4	8	16
RBFA	65%	68.3%	70.18%	86.12%
GLFA	50.82%	52.68%	52.78%	67.54%
LFA	71%	70.47%	66%	61.86%
FF	18.8%	20.6%	21.9%	23.9%

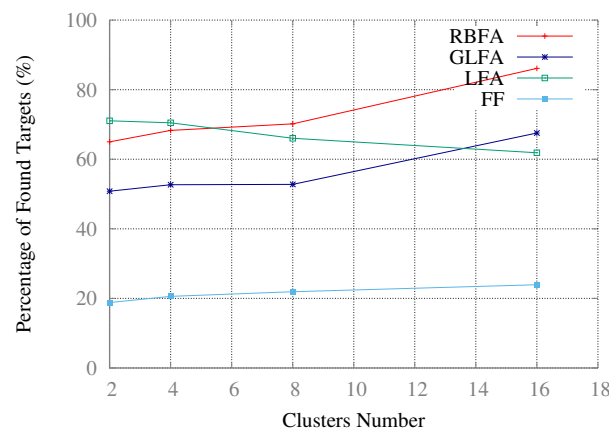


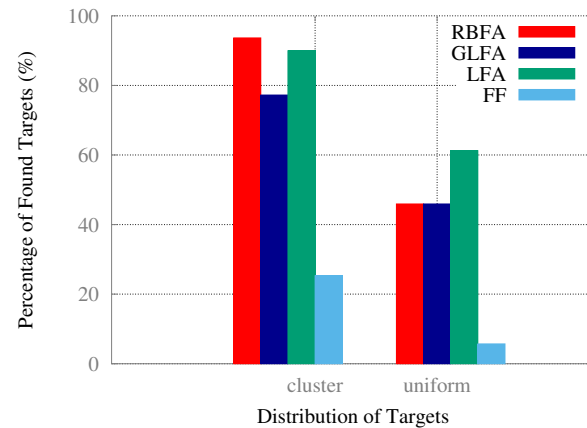
Figure 4. Percentage of found targets when increasing the cluster number—Scenario 3.

4.2.4. Results of Scenario 4

In this scenario, we aim to investigate how different target distribution patterns affect the performance of the four algorithms. We consider two cases: targets grouped into clusters and targets uniformly distributed across the environment. The results are presented in Table 6 and Figure 5. Overall, the performance of all algorithms improves when targets are grouped into clusters. For instance, RBFA achieved 93.66% target discovery, while LFA found 89.9%. Despite both algorithms delivering commendable results, LFA consistently outperforms RBFA in uniformly distributed scenarios. Conversely, RBFA and GLFA exhibit notably poorer performance when targets are uniformly distributed, failing to reach even 50% discovery rates. This suggests that the randomness inherent in RBFA and GLFA strategies may hinder exploration in such cases. Additionally, the results indicate RBFA’s resilience to environmental variations compared to GLFA when targets are clustered. In summary, all discussed algorithms outperform FF in this scenario.

Table 6. Percentage of found targets when changing the distribution of targets (clusters, uniform)—*Scenario 4.*

Clusters Distr.	Clusters	Uniform
RBFA	93.66%	45.93%
GLFA	77.23%	45.93%
LFA	89.9%	61.2%
FF	25.3%	5.7%

**Figure 5.** Percentage of found targets when changing the distribution of targets (clusters, uniform)—*Scenario 4.*

5. Applications and Limitations

This section aims to explore the potentialities and limitations of the proposed algorithm, particularly focusing on the light-based communication mechanism for which we can enumerate several potential real-world applications:

- In underwater exploration scenarios, light presents a viable alternative to traditional radio waves, offering enhanced penetration capabilities in certain conditions;
- Light-based communication is promising for search and rescue missions, providing effective communication among multiple robots operating in disaster-stricken areas;
- Within warehouse environments, where GPS signals may be unreliable, light-based communication could facilitate navigation and coordination among robotic systems;
- Autonomous vehicles, such as drones or cars, stand to benefit from light-based communication, enabling swift transmission of information regarding targets or obstacles;
- In applications requiring secure and reliable communication, such as military operations, industrial settings, and smart healthcare systems, light-based communication offers a compelling alternative.

Despite its potential advantages, light-based communication also entails certain limitations that must be considered for real-world applications in multi-target search tasks:

- Light-based communication necessitates a direct line of sight between communicators, posing challenges in cluttered environments;
- Light has a limited range compared to radio frequency signals and operates effectively only within short distances;
- Environmental conditions, such as fog or rain, can significantly impact the performance of light-based communication, leading to decreased effectiveness;
- Devices utilizing light-based communication require power, necessitating efficient energy consumption management mechanisms for applications with critical battery life requirements.

6. Implementation with Real Robots

This section offers insights into implementing the proposed algorithm with real robots. The design of a multi-target real robot with a light-based communication mechanism should incorporate both traditional robot design aspects (such as autonomous and adaptable navigation, power management, localization, and mapping) and elements specific to our proposition (such as light sensor integration and repulsive behavior). We elaborate on each design aspect below:

1. *Autonomous and adaptable navigation*: the robot should integrate algorithms for planning and executing efficient paths while avoiding obstacles. Additionally, it should be equipped with mobility mechanisms to navigate various ground conditions effectively.
2. *Power management*: design considerations should prioritize energy-efficient mobility mechanisms, supplemented by efficient power management systems to extend operational time.
3. *Localization*: robots should employ localization algorithms to determine their position accurately.
4. *Sensor integration*: implementing a light-based communication system may involve visible light communication or other light modulation techniques. Different colors or frequencies of light could be utilized to encode information, necessitating specific communication protocols to avoid interference.
5. *Repulsive behavior*: algorithms should enable robots to detect the presence of other robots through light signals. Control mechanisms must be implemented to adjust the robot's movements, avoiding the light source and selecting alternative directions for exploration.

7. Conclusions

The utilization of swarm intelligence-based algorithms for collective problem-solving has become pervasive in mobile robotics, encompassing tasks, such as demining, cleaning, search and rescue, among others. Central to these applications is environmental exploration, where multi-robot systems navigate space employing deployment strategies to search for valuable objects. The choice of exploration strategy can range from random to strategic or hybrid walks. Literature suggests that random walking enhances the likelihood of locating objects in unknown environments.

In this study, our focus lies in robot dispersion achieved by implicitly partitioning the environment using light as a repulsive force, inspired by the attraction behavior of fireflies. Specifically, we introduce two algorithms, RBFA and GLFA, which enhance the RB and GL algorithms by integrating light-induced repulsion behavior. These algorithms were implemented in ARGoS to validate their efficacy, and simulation results indicate that RBFA outperforms GLFA.

In future work, we aim to conduct other experiments by considering factors such as variable lighting conditions and interference from external light sources. Moreover, we plan to enhance our light-based communication mechanism to deal with the limited direct line-of-sights. One approach involves enabling robots to relay messages using light-based communication, even without a direct line of sight. Another solution could be integrating alternative communication technologies (such as RFID) to navigate cluttered spaces more effectively. Finally, we intend to perform a wide range of experiments with ARGoS and Gazebo simulators to validate the effectiveness of the proposed algorithm in real-world scenarios.

Author Contributions: Conceptualization, O.Z. and A.G.; formal analysis, O.Z. and A.G., writing—original draft preparation, O.Z. and A.G.; writing—review and editing, O.Z. and A.G.; supervision, G.F. and H.S.; implementation and test: A.B.; funding acquisition, G.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially supported by the European Union—NextGeneration-EU—National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR), Project: “SoBigData.it—Strengthening the Italian RI for Social Mining and Big Data Analytics”, Prot. IR0000013—Avviso n. 3264 del 28/12/2021.

Data Availability Statement: Data are contained within the article.

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

References

1. Sahin, E. *Swarm Robotics*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3342, pp. 10–20. [[CrossRef](#)]
2. Şahin, E.; Girgin, S.; Bayindir, L.; Turgut, A.E. Swarm Robotics. In *Swarm Intelligence*; Natural Computing Series; Springer: Berlin/Heidelberg, Germany, 2008; pp. 87–100. [[CrossRef](#)]
3. Zedadra, O.; Guerrieri, A.; Seridi, H. LFA: A Lévy walk and firefly-based search algorithm: Application to multi-target search and multi-robot foraging. *Big Data Cogn. Comput.* **2022**, *6*, 22. [[CrossRef](#)]
4. Yang, X.S. Firefly Algorithms for Multimodal Optimization. In *Stochastic Algorithms: Foundations and Applications*; Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer: Berlin/Heidelberg, Germany, 2009; Volume 5792 LNCS, pp. 169–178. [[CrossRef](#)]
5. Benavides, F.; Monzón, P.; Chanel, C.P.C.; Grampín, E. Multi-robot Cooperative Systems for Exploration: Advances in dealing with constrained communication environments. In Proceedings of the 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), Recife, Brazil, 8–12 October 2016; pp. 181–186.
6. Palmieri, N.; De Rango, F.; She Yang, X.; Marano, S. Multi-robot cooperative tasks using combined nature-inspired techniques. In Proceedings of the 7th International Joint Conference on Computational Intelligence, IJCCI 2015, Lisbon, Portugal, 12–14 November 2015; Volume 1, pp. 74–82. [[CrossRef](#)]
7. Dimidov, C.; Oriolo, G.; Trianni, V. Random walks in swarm robotics: An experiment with Kilobots. *Lect. Notes Comput. Sci.* **2016**, *9882*, 185–196. [[CrossRef](#)]
8. Palmieri, N.; Marano, S. Discrete firefly algorithm for recruiting task in a swarm of robots. In *Nature-Inspired Computation in Engineering*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 133–150.
9. Katada, Y.; Nishiguchi, A.; Moriwaki, K.; Watakabe, R. Swarm robotic network using Lévy flight in target detection problem. *Artif. Life Robot.* **2016**, *21*, 295–301. [[CrossRef](#)]
10. Palmieri, N.; Yang, X.S.; De Rango, F.; Marano, S. Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption. *Neural Comput. Appl.* **2019**, *31*, 263–286. [[CrossRef](#)]
11. Palmieri, N.; Rango, F.D.; Yang, X.S.; Marano, S. Bio-inspired strategies for the coordination of a swarm of robots in an unknown area. *Stud. Comput. Intell.* **2017**, *669*, 96–112. [[CrossRef](#)]
12. Suárez, P.; Iglesias, A. Bat algorithm for coordinated exploration in swarm robotics. *Adv. Intell. Syst. Comput.* **2017**, *514*, 134–144. [[CrossRef](#)]
13. Santos, R.G.; De Freitas, E.P.; Cheng, S.; De Almeida Ribeiro, P.R.; Muniz De Oliveira, A.C. Autonomous Exploration Guided by Optimisation Metaheuristic. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018, Singapore, 18–21 November 2018; pp. 1759–1764. [[CrossRef](#)]
14. Abuomar, L.; Al-Aubidy, K. Cooperative search and rescue with swarm of robots using binary dragonfly algorithm. In Proceedings of the 2018 15th International Multi-Conference on Systems, Signals and Devices, SSD 2018, Yasmine Hammamet, Tunisia, 19–22 March 2018; pp. 653–659. [[CrossRef](#)]
15. De Rango, F.; Palmieri, N.; Yang, X.S.; Marano, S. Swarm robotics in wireless distributed protocol design for coordinating robots involved in cooperative tasks. *Soft Comput.* **2018**, *22*, 4251–4266. [[CrossRef](#)]
16. Khaluf, Y.; Van Havermaet, S.; Simoens, P. *Collective lévy walk for efficient exploration in unknown environments*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11089, pp. 260–264. [[CrossRef](#)]
17. Huang, X.; Arvin, F.; West, C.; Watson, S.; Lennox, B. Exploration in Extreme Environments with Swarm Robotic System. In Proceedings of the 2019 IEEE International Conference on Mechatronics, ICM 2019, Ilmenau, Germany, 18–20 March 2019; Volume 1, pp. 193–198. [[CrossRef](#)]
18. Pang, B.; Song, Y.; Zhang, C.; Wang, H.; Yang, R. A Swarm Robotic Exploration Strategy Based on an Improved Random Walk Method. *J. Robot.* **2019**, *2019*, 6914212. [[CrossRef](#)]
19. 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 Pap.* **2021**, *54*, 1–6. [[CrossRef](#)]
20. Garg, V.; Tiwari, R.; Shukla, A. Comparative Analysis of Fruit Fly-Inspired Multi-Robot Cooperative Algorithm for Target Search and Rescue. In Proceedings of the 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 17–19 June 2022; pp. 444–450.
21. Yang, L.; Xing, B.; Li, C.; Wang, W. Research on Artificial Bee Colony Method Based Complete Coverage Path Planning Algorithm for Search and Rescue Robot. In Proceedings of the 2022 5th International Symposium on Autonomous Systems (ISAS), Hangzhou, China, 8–10 April 2022; pp. 1–6.

22. Fricke, G.M.; Hecker, J.P.; Griego, A.D.; Tran, L.T.; Moses, M.E. A distributed deterministic spiral search algorithm for swarms. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Daejeon, Republic of Korea, 9–14 October 2016; pp. 4430–4436. [[CrossRef](#)]
23. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36–50. [[CrossRef](#)]
24. Isaacs, J.T.; Dolan-Stern, N.; Getzinger, M.; Warner, E.; Venegas, A.; Sanchez, A. Central place foraging: Delivery lanes, recruitment and site fidelity. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020; pp. 319–324.
25. Ousingsawat, J.; Earl, M.G. Modified lawn-mower search pattern for areas comprised of weighted regions. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 918–923.
26. Pinciroli, C.; Trianni, V.; O’Grady, R.; Pini, G.; Brutschy, A.; Brambilla, M.; Mathews, N.; Ferrante, E.; Di Caro, G.; Ducatelle, F.; et al. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* **2012**, *6*, 271–295. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.