

Article

LFA: A Lévy Walk and Firefly-Based Search Algorithm: Application to Multi-Target Search and Multi-Robot Foraging

Ouarda Zedadra ^{1,*} , Antonio Guerrieri ²  and Hamid Seridi ¹ 

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

² Institute for High Performance Computing and Networking (ICAR), CNR—National Research Council of Italy, Via P. Bucci 8/9C, 87036 Rende, Italy; antonio.guerrieri@icar.cnr.it

* Correspondence: zedadra.ouarda@univ-guelma.dz

Abstract: In the literature, several exploration algorithms have been proposed so far. Among these, Lévy walk is commonly used since it is proved to be more efficient than the simple random-walk exploration. It is beneficial when targets are sparsely distributed in the search space. However, due to its super-diffusive behavior, some tuning is needed to improve its performance, specifically when targets are clustered. Firefly algorithm is a swarm intelligence-based algorithm useful for intensive search, but its exploration rate is very limited. An efficient and reliable search could be attained by combining the two algorithms since the first one allows exploration space, and the second one encourages its exploitation. In this paper, we propose a swarm intelligence-based search algorithm called Lévy walk and Firefly-based Algorithm (LFA), which is a hybridization of the two aforementioned algorithms. The algorithm is applied to Multi-Target Search and Multi-Robot Foraging. Numerical experiments to test the performances are conducted on the robotic simulator ARGoS. A comparison with the original firefly algorithm proves the goodness of our contribution.

Keywords: swarm intelligence; swarm robotics; Lévy walk; Firefly algorithm; LFA; multi-target search (MTS); Multi-Robot Foraging (MRF)



Citation: 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. <https://doi.org/10.3390/bdcc6010022>

Academic Editor: Moulay A. Akhloufi

Received: 16 December 2021

Accepted: 17 February 2022

Published: 21 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

In the future, Multi-Robots Systems (MRS) are intended to replace humans in some tasks, such as exploration, search and rescue, monitoring, surveillance, cleaning, and so on. To accomplish such tasks, mechanisms of distributed coordination and cooperation are desirable for MRS. Swarm robotics is a new approach to the coordination of large numbers of robots whose main inspiration stems from the observation of social insects [1]. Swarm robotics has recently come out as the realizations of the swarm intelligence (SI) toward the MRS. It emphasizes the incorporation of individuals and lifelike interactions, both among individuals and between the individuals and their environment. Swarm robotics exposes a synchronized behavior at the system level which emerges despite (i) the limited capabilities of the individuals, (ii) the absence of centralized coordination, and (iii) the easiness of interactions [2].

Firefly algorithm (FA) is an SI-based algorithm that takes advantage of lights to coordinate fireflies, which, in nature, emit bioluminescent light to allure their mates or preys. More is the brightness they emit, more is the attraction that they have. The light intensity is proportional to the associated luminescence quantity called luciferin, which attracts other fireflies within their neighborhood. The standard FA employs a full attraction model, which results in oscillations during the search process, and low attraction can also lead to premature convergence. Therefore, the number of attractions is very important [3]. The standard Firefly algorithm adopts a fixed step length which can cause it to become trapped in the local optima, thus causing low precision. The exploration rate of FA is very limited. FA is useful for MRS since its real application is guaranteed by Ambient Light

Sensors (ALS) technology. In this case, it is important to use sensors on each robot, not only in order to help it to become directionally oriented but also to follow the light emitted by the robot chief or to help to find the goal easier [4].

Lévy walks (LW) are the most efficient random walk strategies known to efficiently search spaces when targets are sparsely distributed. Such search patterns have a characteristic length scale. They are characterized by a set of small steps connected by a long step. The Lévy steps cover more than the Gaussian model using the same length of time and number of steps. In addition, the length of steps in the Lévy model is very long compared to the Gaussian model, which makes it possible to explore new regions in the environment, cover more space, and avoid local and repetitive search.

In this work, we consider multi-robot exploration and foraging in an unknown environment. In order to benefit from the ALS technology, we propose in this paper a Lévy walk and Firefly-based search algorithm. These two algorithms can be seen as complementary, since LW provides global search and FA permits local search. With a fine-tuning between the two algorithms, an efficient and reliable search of the work space could be reached. The proposed algorithm is used to (1) decrease the super-diffusive behavior of LW by using FA for exploitation, (2) increase the exploration rate of FA by using LW, and (3) obtain an equilibrium between exploration and exploitation. The algorithm will be applied to Multi-Target Search (MTS) and Multi-Robot Foraging (MRF) problems, and the performances will be tested numerically through the ARGoS simulator. It is worth noting that several improvements have been already achieved on the LW algorithm in our previous work in Reference [5].

The rest of the paper is structured as follows. In Section 2, we will introduce the original LW and FAs. In Section 3, we will show some related works. In Section 4, we will present the Finite State Machine (FSM) of the proposed LFA and its pseudo code with the description of all the involved parameters. Section 5 will provide the experimental analysis of LFA and its comparison with the FA. In Section 6, we will apply the LFA to the Multi-Robot Foraging problem, and we will present some experiments with a discussion about results and comparisons. Finally, we will draw out some conclusions and highlight some perspectives in Section 7.

2. Lévy Walk and Firefly Algorithms

In this section, we will introduce the original LW algorithm and talk about the FA and its characteristics.

LW is a random walk whose step size varies according to a Lévy probability distribution. It is useful for the simulated environment where targets are distributed sparsely and randomly. Lévy distribution for a step size can be approximated as in Equation (1) [6].

$$Sl = l_0 \times \left(\frac{1}{U^{\frac{1}{\beta}}} - 1 \right), \quad (1)$$

where U is uniformly distributed in the interval $[0, 1]$, and l_0 and β are parameters to be tuned in order to best fit a given landscape. l_0 and Sl represent, respectively, a scale variable and the step length.

FA was first developed by Xin-She Yang [7]. It was inspired by the flashing patterns of fireflies which are used to attract mating partners and to attract potential prey. It is very efficient in dealing with multi-modal, global optimization, multidimensional, and nonlinear problems [8]. In FA, the two most important issues are the variation of light intensity and the formulation of attractiveness. FA uses the following idealized rules [9]:

- Fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex.
- The attractiveness is proportional to the brightness. Moreover, it lowers when the distance increases. Thus, given two flashing fireflies, the one that is less bright will

move in the direction of the brighter one. A firefly moves randomly until a brighter firefly is located.

- The landscape of the objective function determines the brightness of a firefly.

As a firefly's attractiveness is proportional to the light intensity seen by the adjacent fireflies, the variation of attractiveness β with the distance r is represented in Equation (2) [7]:

$$\beta = \beta_0 \times e^{-\gamma \times r^2}. \quad (2)$$

The movement of firefly i (at position x_i) that is attracted by another more attractive firefly j (at position x_j) is determined by Equation (3) [10]:

$$x_i^{t+1} = x_i^t + \beta_0 \times e^{-\gamma \times r_{ij}^2} \times (x_j^t - x_i^t) + \alpha \times \zeta_i^t, \quad (3)$$

where β_0 , γ , r_{ij} , α , and ζ_i^t represent, respectively, the attractiveness in distance r , an absorbent coefficient, the distance between x_i and x_j , a control parameter, and a random variable.

The main advantages of the FA are [10]:

- Automatic partition of the population into subgroups so that each subgroup can swarm around a local mode. Thus, FA can deal with multi-modal optimization;
- The attraction mechanism of the FA can speed up the convergence. It is nonlinear and, thus, may be richer in terms of dynamic characteristics;
- FA can efficiently deal with a variegated range of optimization problems since it can include Particle Swarm Optimization, Differential Evolution, and Simulated Annealing with its special cases.

3. Related Works

Lévy walk and Firefly algorithms have been used in several fields so far. In Section 3.1, we will analyze their use for the Robotic Search Problem. Then, we present a simple discussion of related works in Section 3.2.

3.1. Lévy Walk and Firefly-Based Search Works

A Lévy looped search algorithm to locate mobile targets with a swarm of non-interacting robots was proposed by Lenagh and Dasgupta [11]. In their proposal, each searcher returns to its initial position by replacing straight ballistic segments with loops. The length of each loop was sampled through a power-law distribution. The authors of Reference [12] enhance the Lévy flight to one moving target positioned according to a radial probability density function, considering also the detection range, i.e., the resolution of an onboard sensor. In this paper, some simulations endorse the better performance of Lévy flight with respect to the random walk when the search is in the intermediate range of the sensor resolution.

Reference [13] introduces a three-layered subsumption architecture with the aim of analyzing the effect of the step size of the random walk and the number of robots on a target detection problem. When a target is found, the robots send this information to a base station through intermediate robots by using multi-hop communication. The authors also performed several real experiments confirming that a variable step size according to a Lévy probability distribution is useful for an exploration strategy. Some other work on Lévy walk may be taken from the survey in Reference [14]. Here, the authors outline the state of the art in the field and define further perspectives. Even if Lévy walk is a valuable and pretty much used algorithm, it presents some limitations regarding the time spent in changing frequently the search area. It needs some tuning of parameters to produce a balance between exploration and exploitation.

The authors of Reference [15] propose a Discrete Firefly algorithm (DFA) for mine disarming tasks in an unknown area. The introduced algorithm mixes the foraging behavior of colonies of ants [16] and the Firefly algorithm. The first one is useful to disseminate robots

so to be able to explore different regions, and the latter is used when a mine is detected to attract other robots in the neighborhood to collaborate in the mine disarming. Robots utilize direct broadcast communication to communicate to neighbors in their wireless range the place in which the mine is detected. A multi-robot searching algorithm based on a combination of a Lévy walk and an artificial potential field inducing repulsion among robots was proposed and tested in Reference [17]. Results for the cases, in which up to twenty robots are used, showed that the repulsion increases search efficiency in terms of search time. Sutanty et al. [18] proposed a bio-inspired algorithm for underwater search scenarios. They enhanced Lévy flight performances by introducing an adaptation strategy based on the Firefly Optimization algorithm (FO). The FO-based adaptation strategy is based on two rules: (1) update attractiveness: the robot increases attractiveness whenever a target is located, applying linear decay when no target is located, and (2) communication: a robot communicates and follows the robot in its sensing range with attractiveness higher to its own. Group Lévy foraging with an artificial pheromone communication between robots was studied by Fujisawa and Dobata [19]. Here, each robot had a tank filled with a pheromone (alcohol) which was sprayed around by a micro-pump. Rovers also carried alcohol and touch sensors, and their motion was controlled by a program that took into account the local pheromone concentration.

LW algorithm and FA have been also used for Numerical Optimization. Yang and Deb proposed, in Reference [20], a hybrid optimization algorithm called Eagle Strategy. The latter used the Lévy walk and Firefly for stochastic optimization. The eagle performed the Lévy walk in the whole search domain. When a prey was located, it switched to local search using the standard Firefly algorithm [7]. Eagle Strategy efficiency was tested on various mathematical functions. To increase the exploration and improve the global search of the FA, Hassanzadeh and Kanan [21] proposed a fuzzy-based FA. Yu et al. [22] proposed a Firefly algorithm which introduces a variable step size to the standard FA with the aim of solving numerical problems. The proposed algorithm uses a dynamic strategy to adjust the step α in the search process, to ensure a balance between exploration and exploitation. Experiments were conducted on benchmark functions. Wang et al. [3] proposed a FA with neighborhood attraction model inspired by the *k-neighborhood* concept to overcome the oscillations during the search process and high computational time complexity [3]. Each firefly is attracted only by brighter fireflies in a predefined neighborhood, rather than those from the whole population. Several experiments were also conducted on benchmark functions that highlighted the performances of the proposed algorithm.

3.2. Related Works Discussion

According to the related work that has been presented in the previous sub-section, only the works in Reference [18,20] realize a hybridization of LW algorithm and FA. In Reference [18], the random motion element of the FA is replaced with a random number from the Lévy distribution generator. It was applied to the multi-robot exploration problem in an underwater environment. In Reference [20], LW is used in global search and FA for local search. Here, it was applied to stochastic optimization. In the same direction, we propose in this work a hybridization between the two aforementioned algorithms. LW is used in global search and in local search, according to some triggering conditions, while FA is just used to attract other robots in the neighborhood. This hybridization is used in multi-targets search and Multi-Robot Foraging with walking robots.

4. The Proposed Lévy Walk and Firefly-Based Search Algorithm (LFA)

In this section, we describe the proposed Lévy walk and Firefly-Based Search Algorithm (LFA). In such an algorithm, we use a three-stage search strategy: (1) global search using LW with $\beta = 0.75$ for large step size, (2) local search using LW with $\beta = 1.25$ for small step size, and (3) attraction model using FA.

In Section 4.1, we detail the parameters of the LFA. Then, we present the Finite State Machine (FSM) and the pseudo-code of the proposed Firefly-Based Search algorithm (LFA) in Section 4.2.

4.1. The Parameters of the LFA

We consider in the LFA the following assumptions:

- the attractiveness is either $\beta_0 = 0$ or $\beta_0 = 1$. β_0 is the *attraction*;
- the Light Intensity I is diffused whenever $\beta_0 = 1$; and
- since the attractiveness is relative, it should be seen in the beholder or judged by other fireflies. We represent the light intensity through a color to be seen by the other robots.

We then consider the following rules:

- *Random walk Management*: The random steps are generated using Lévy walk according to Equation (1) [6]. I_0 is fixed to 1 as in Reference [6]. After several experiments, β is fixed to 0.75 in the *exploration state* in order to increase the probability of generating steps with long size, while the same parameter is fixed to 1.25 in *exploitation state* to obtain steps with small size.

- *Light Intensity Management*: Light Intensity I is managed using these three rules:

1. Increased each time a target is found according to Equation (4). I_t represents light intensity at step t , and ρ is the time spent in the exploitation state:

$$I_{t+1} = (1 + \rho) \times I_t. \quad (4)$$

2. Decreased each time a target is not found according to Equation (5), where ρ represents the time spent in the exploitation state:

$$I_{t+1} = (1 - \rho) \times I_t. \quad (5)$$

3. Diffused regarding distance in each step according to Equation (6), where I_r represents the light intensity at distance r , μ is a random real value $\in [0, 1]$, and r is a predefined distance:

$$I_r = \frac{I_0}{1 + \mu \times r^2}. \quad (6)$$

- *Following Behavior Management*: To follow a detected light, the robot uses the movement rule according to Equation (7). The second term of Equation (7) is due to the attraction. The robot keeps moving until the criteria in Equation (8) is verified. In the equations, x_i , x_j , y_i , and y_j represent the coordinates (in the Cartesian axes) of *Robot_i* and *Robot_j*, r_{ij} stands for the distance between *Robot_i* and *Robot_j*, D is the distance to be kept between *Robot_i* and *Robot_j* in *Follow Light state* (see Algorithm 3 in the following), and s_r and f_r represent, respectively, the sensing and the following ranges.

$$x_{i+1} = x_i + \beta_0 \times e^{-\mu \times r_{ij}^2} \times (x_j - x_i), \quad (7)$$

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} = D, \quad (8)$$

$$D = S_r \times f_r. \quad (9)$$

- *Robot's Velocity Management*: The velocity of robots is increased when a light is detected, according to Equation (10), and it is decreased once the criteria in Equation (8) is verified, according to Equation (11). v_t and ω_i represent, respectively, the robots' velocity and an inertia coefficient.

$$v_{i+1} = 2 \times (v_t \times \omega_i), \quad (10)$$

$$v_{i+1} = \frac{1}{2} \times (v_t \times \omega_i). \quad (11)$$

Some technical requirements are clarified below:

- According to Yang and Deb [20], we initialize $\beta_0 = 1$ and $\mu = 1$.
- The sensing range of S_r is fixed to 0.3 m.
- To fulfill the technical specification of the Firefly algorithm that specifies a longer range for communication than for sensing (*Intensity range*, or $r > S_r$), we used Equation (4) to increase light intensity (light intensity is increased by ρ percentage, which represents the percentage of elapsed time in the exploitation state T_{max}). Decreasing the intensity follows the same principle as increasing it; the intensity is decreased with ρ percentage, which represents the percentage of elapsed time in exploitation state T_{max} . Then, we specify a maximum *Intensity range* to allow the subdivision of swarm into sub-groups.
- In order to subdivide the search region between firefly robot and follower robots, we fixed the following distance to D in Equation (9). As shown in Figure 1, D represents the third S_r of Firefly robot, which means that the first and the second S_r are areas to be explored by the firefly robot, while the third one is allocated to follower robots.

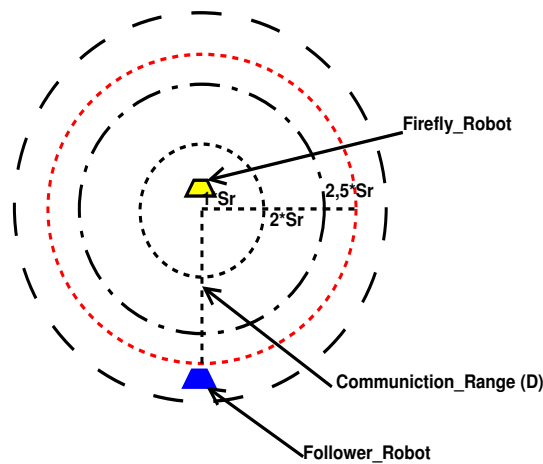


Figure 1. Defined ranges for sensing (range in which the robot can directly localize targets and obstacles, where $1 \times S_r$ is the sensing range and $2 \times S_r$ is twice the sensing range) and communication (distance to be kept between *follow_robot* and *Firefly_robot*).

4.2. The Finite State Machine of LFA Robots

Robots can switch between the four states: *Exploration*, *Exploitation*, *Following*, or *Obstacle Avoidance* according to their sensor inputs. The FSM of LFA robots behavior is given in Figure 2.

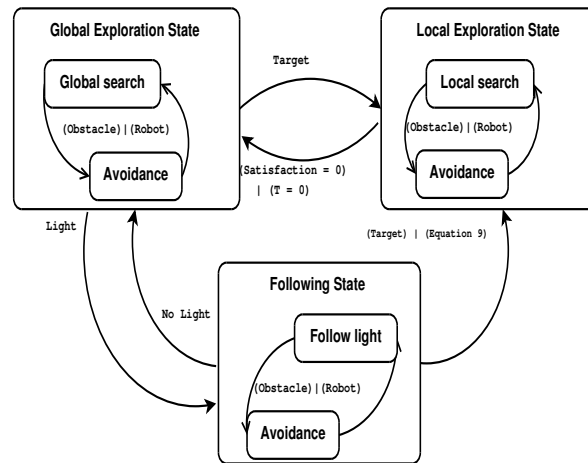


Figure 2. FSM of LFA Robots, where T is the time fixed for exploitation.

The following Algorithms 1–4, represent, respectively, the behavior of robots in the states: *Global Exploration*, *Local Exploration*, *Follow Light*, and *Obstacle Avoidance*.

In *Global Exploration*, the robots use the LW algorithm with $\beta = 0.75$ to generate long steps length. Whenever a target is located, the robot changes its state to *Local Exploration* and uses $\beta = 1.25$ to generate small steps lengths in order to locally explore the area if it is rich with targets. If a light is detected, a robot follows it to enter in the rich area. Finally, when an obstacle is detected, the robot changes to *Avoidance* state.

Algorithm 1: Global Exploration (exploration).

```

1 repeat
2   if (global step list = ∅) then
3     Generate a new global step list using Equation (1) with  $\beta = 0.75$ ;
4   else
5     Move using the first list step and update global step list;
6 until  $(\exists \text{ Obstacle}) \vee (\exists \text{ Target}) \vee (\text{Step terminated})$ ;
7 if  $(\exists \text{ Obstacle})$  then
8   Goto Avoidance (Algorithm 4);
9 else
10  if  $(\exists \text{ Target})$  then
11    Goto Local Exploration (Algorithm 2);
12  else
13    Goto Follow light (Algorithm 3);

```

Algorithm 2: Local Exploration (exploitation).

```

1 repeat
2   if (local step list = ∅) then
3     Generate a new local step list using Equation (1) with  $\beta = 1.25$ ;
4   else
5     Move using the first list step and update local step list;
6 until  $((T_{max} \neq 0) \wedge (\text{satisfaction} \neq 0))$ ;
7 if  $(\exists \text{ Target})$  then
8   Increase light intensity using Equation (4);
9   Diffuse light using Equation (6);
10  Update satisfaction and  $T_{max}$ ;
11 else
12   if  $(\exists \text{ Obstacle})$  then
13     Goto Avoidance (Algorithm 4);
14 Goto Global Exploration (Algorithm 1);

```

Algorithm 3: Follow light.

```

1 repeat
2   Increase velocity using Equation (10) and walk toward the light;
3   if ( $\exists$  Target) then
4     Goto Local Exploration (Algorithm 2);
5   else
6     if ( $\exists$  Obstacle) then
7       Goto Avoidance (Algorithm 4);
8 until  $((D \text{ attended}) \wedge (\nexists \text{ light}))$ ;
9 if ( $D \text{ attended}$ ) then
10  Goto Local Exploration (Algorithm 2);
11 else
12  if ( $\nexists \text{ light}$ ) then
13    Goto Global Exploration (Algorithm 1);

```

Algorithm 4: Obstacle Avoidance.

```

1 Collect the proximity sensors' reading values;
2 Calculate the maximal value  $V_{max}$ ;
3 if ( $V > 0$ ) then
4   Calculate the angle  $A$  of  $V_{max}$ ;
5   if (Obstacle in front of robot) then
6     Walk backward;
7   else
8     if (Obstacle in the left side of the robot) then
9       Move right;
10    else
11      Move left;

```

5. Experimental Analysis

In this study, we focus on the search of multiple targets by a swarm of nb_R robots. The experimental simulation has been implemented in the multi-physics robot simulator ARGoS [23], which is able to simulate efficiently large-scale swarms of robots of any kind. We have performed several computer simulations to test the performance of the proposed algorithm varying several criteria (i.e., number of robots, environment size, target distribution—uniform or clustered—and number of clusters).

Table 1 shows the considered simulation scenarios. Each simulation has been run for 15 min. In all the simulations, the obstacle density is 10%. As these are stochastic algorithms, the data presented here represent 50 simulations where the Standard Deviation (STD) is reported to ensure the probabilistic quality of results. In the four realized scenarios, we have used the performance metric *number of targets found*. Scenarios 1, 2, 3, and 4 have been used to test, respectively, the influence given by the variation of: (i) the number of robots, (ii) the environment size, (iii) the number of clusters, and (iv) the distribution of targets (clustered or uniform). The parameters of each scenario are depicted in Table 1.

Table 1. Simulation scenarios parameters.

Parameter	Value
Varying the number of robots—Scenario 1	
Number of Robots	50, 100, 200, 400, 800, 1600
Target Distribution	clustered
Number of Clusters	10
Number of Targets	2000
Environment Size	240×240

Table 1. Cont.

Parameter	Value
Varying the environment size—Scenario 2	
Environment Size	$80 \times 80, 160 \times 160, 320 \times 320, 640 \times 640$
Number of Robots	80
Target Distribution	clustered
Number of Targets	2000
Number of Clusters	10
Varying the number of clusters—Scenario 3	
Number of Clusters	2, 4, 8, 16
Number of Robots	80
Target Distribution	clustered
Number of Targets	2000
Environment Size	120×120
Varying the distribution of targets—Scenario 4	
Target Distribution	clustered, uniform
Number of Robots	80
Number of Targets	800
Number of Clusters	10
Environment Size	120×120

5.1. Results and Discussion

The performances of the LFA have been compared with the performances obtained by the original FireFly algorithm (FF). Below, we present the results obtained by the two algorithms in the four described scenarios.

5.1.1. Results of Scenario 1

The aim of this scenario is that of studying the influence given by the number of robots on performances. For this purpose, we have increased the number of robots from 50 to 1600. Table 2 and Figure 3a show, respectively, the results obtained by LFA and FF algorithm. The total number of targets found increases when increasing the number of robots. The LFA gives better results than the FF one. In Table 2, we can observe that the LFA with 1600 robots reaches all the targets. With the local search provided by LW, a robot can explore the neighborhood of an area if it is rich in targets. In addition, with the attraction behavior, other robots can join a firefly robot to exploit the rich area, rather than consuming time in global exploration.

Table 2. Targets found when increasing the number of robots—Scenario 1.

Number of Robots	50	100	200	400	800	1600
LFA	671.2	1182.6	1545.5	1946.8	1997.6	2000
FF	53.8	186.6	364.2	546.4	993.9	1209.3

5.1.2. Results of Scenario 2

This scenario wants to study the influence of the environment size on the performances of the LFA and FF algorithm. In this case, we have varied the environment size from 80×80 to 160×160 , 320×320 , and 640×640 . Table 3 and Figure 3b show the obtained results. In both the algorithms, the Total number of targets found decreases when increasing the environment size. This can be explained because, by increasing the environment size, the area to be explored is also increased and, as a consequence, so is the exploration time. In addition, in this case, the LFA gives the best results. In LFA, the use of global exploration with LW can increase the probability of visiting new regions, hence increasing the probability of locating new targets. In addition, the flashing behavior of robots can be beneficial in increasing the total number of targets found. However, the random walk used in FF decreases the probability of exploring new regions.

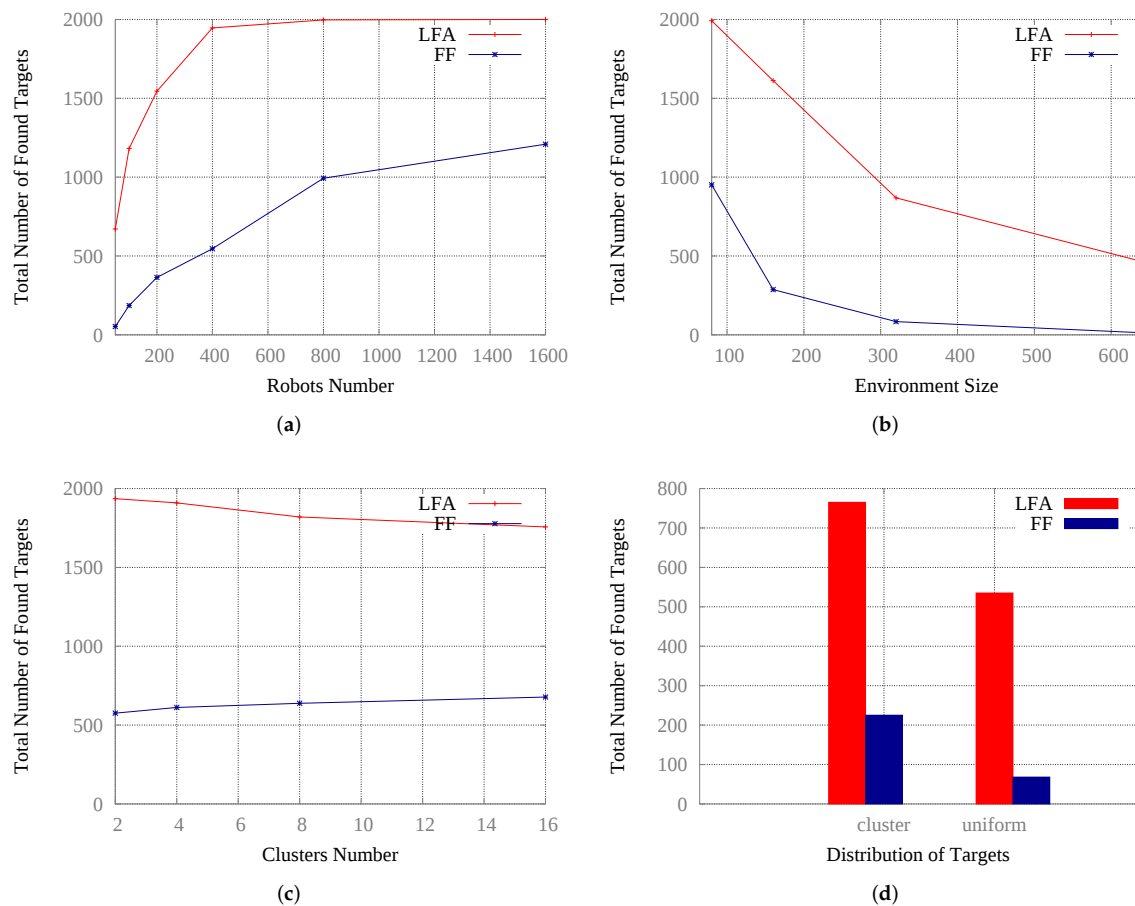


Figure 3. Simulation results: (a) results of Scenario 1, (b) results of Scenario 2, (c) results of Scenario 3, (d) results of Scenario 4.

Table 3. Targets found when increasing the environment size—*Scenario 2*.

Environment Size	80 × 80	160 × 160	320 × 320	640 × 640
LFA	1992.3	1612.6	869.6	465.9
FF	951.2	287.1	84.1	13.2

5.1.3. Results of Scenario 3

The purpose of this scenario is to study how performances are affected by the variation in the *number of clusters*. To reach this goal, we have aggregated targets in clusters of different sizes (i.e., 2, 4, 8, and 16). Table 4 and Figure 3c show the results obtained by the LFA and FF algorithm. In particular, we can observe a slight degradation in the *total number of targets found* in the LFA when the clusters augment. Vice-versa, with the FF, we have an increment on the targets found by augmenting the number of clusters. However, the results we obtain with the LFA are always better than the ones we have with the FF.

Table 4. Targets found when increasing the number of clusters—*Scenario 3*.

Number of Clusters	2	4	8	16
LFA	1936.9	1909.4	1820.8	1757.2
FF	576.1	612	638.5	678.7

5.1.4. Results of Scenario 4

The influence of the target distribution is studied in this last scenario. The distribution can be a cluster that contains a limited number of targets or a uniform distribution. Table 5 and Figure 3d show, respectively, the results obtained by the LFA and FF algorithm. It is worth noting that the *total number of targets found* increases in the two algorithms when changing the targets distribution from *uniform* to *clustered*. In addition, in this case, the results of the LFA give better values with respect to the FF algorithm. The fact that FF gives the worst results when the distribution is uniform proves that the search strategy in FF has a negative influence on global exploration. FF algorithm adopts a Gaussian distribution in its random search, causing steps to have the same length in most of the iterations. However, the LFA gives better results if targets are *uniformly distributed* due to the LW strategy that allows robots to explore different areas at each step. The obtained results show that the LFA is robust to variations, such as target distribution, due to the decentralized coordination and the distribution of perception presented by LFA robots.

Table 5. Targets found when changing the targets distribution (clustered, uniform)—Scenario 4.

Distribution of Targets	Clusters	Uniform Distribution
LFA	765.1	535.5
FF	225.1	68.6

6. Application of LFA to Multi-Robot Foraging Problem (LFFA Algorithm)

In this section, we generalize the LFA to the Multi-Robot Foraging problem (LFFA). In particular, in Section 6.1, we present the FSM of the LFFA foragers, and, in Section 6.2, we show the realized experimental scenarios and comparisons with the FF algorithm enhanced with the same homing behavior of LFFA, to test the LFFA performances.

6.1. FSM of LFA Foragers

Below, we present the FSM of the LFA foragers (Figure 4). A quick explanation of the behaviors follows, while a detailed one was already given in Reference [24]. Robots use six elementary states: *Global search* (in the composite state *Global exploration*), *Local search* (in the composite state *Local exploration*), *Follow robot light* (in the composite state *Following*), *Follow home light* (in the composite state *Homing*), *Return to Cluster* (in the composite state *Return to cluster*), and *Avoidance*, which is accessible from all the states whenever an obstacle or robot is found.

6.2. Experimental Scenarios

In Reference [24], we already presented some preliminary experiments to study the performances of LFA in foraging problem. Table 6 illustrates the simulation scenarios we have realized. Each simulation has been run for 15 min, and the obstacle density in all the simulations has been set at 10%. The results shown represent 50 simulation executions. In Scenario 1, we have increased the number of robots from 800 to 6400 (i.e., 800, 1600, 3200, and 6400). In Table 7 and Figure 5a, we report the obtained foraging time for all the cases taken in consideration. In Scenario 2, we increased robot capacity from 1 to 16 (i.e., 1, 4, 8, and 16). In Table 8 and Figure 5b, we show the total number of collected targets for all the cases taken in consideration. For *Homing* behavior, robots follow light emitted by the base, and, for the *Return to Cluster*, robots follow the direction of the cluster stored before depositing.

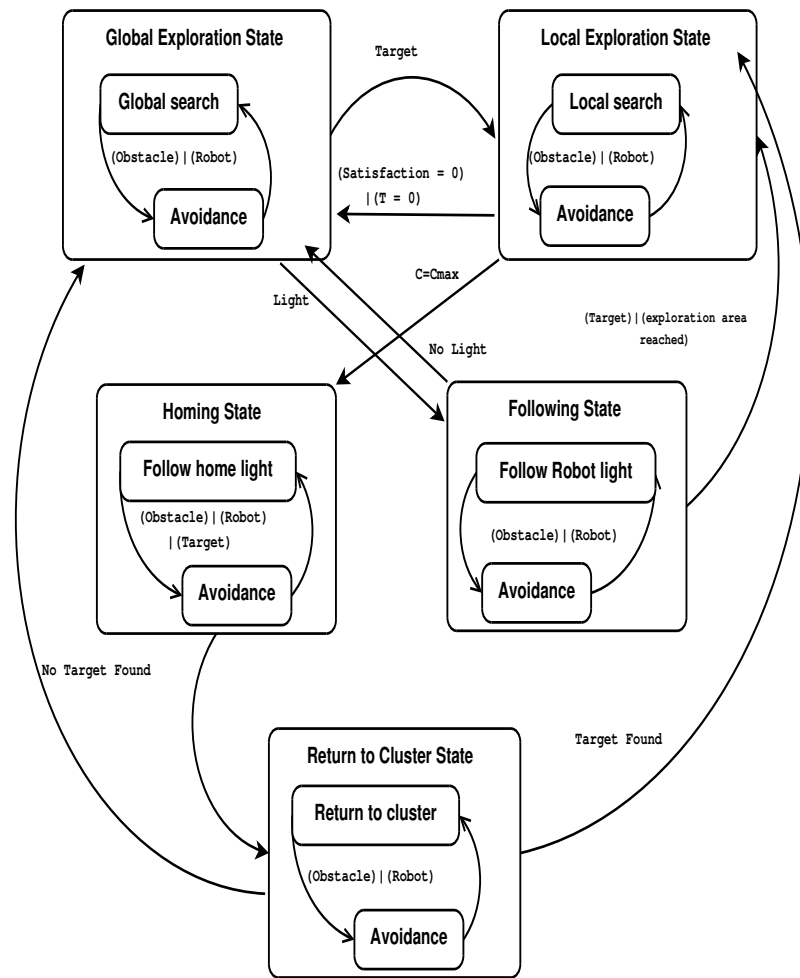


Figure 4. FSM of LFFA Robots.

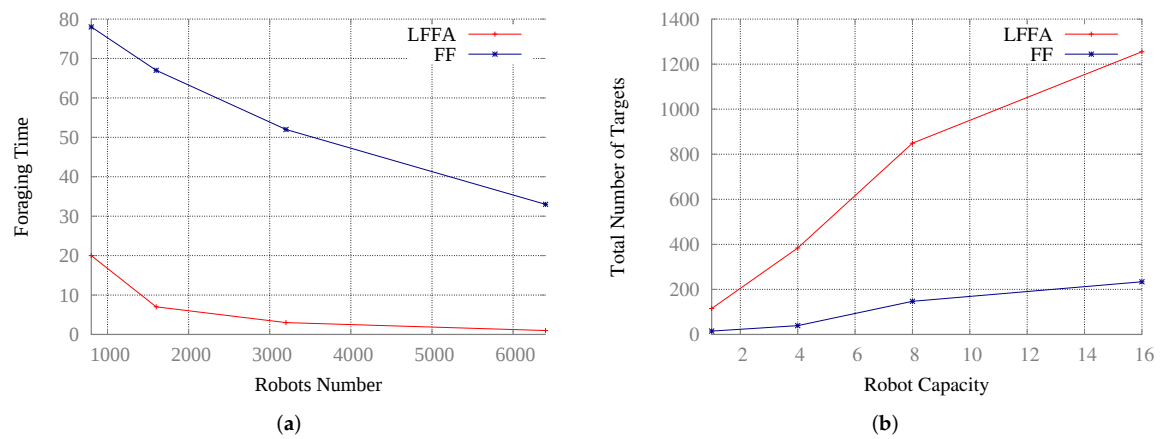


Figure 5. Simulation results: (a) foraging time in Scenario 1, (b) number of collected food in Scenario 2.

Table 6. Simulation scenarios parameters.

Parameter	Value
Varying the number of robots—Scenario 1	
Number of Robots	800–6400
Target Distribution	clustered
Number of Targets	1000
Number of Clusters	10
Environment Size	240 m × 240 m
Varying the capacity of robots—Scenario 2	
Number of Robots	80
Capacity of the Robots	1–16
Target Distribution	clustered
Number of Targets	2000
Number of Clusters	10
Environment Size	120 m × 120 m

Table 7. Foraging time when varying the number of robot—Scenario 1.

Number of Robots	800	1600	3200	6400
LFFA—Foraging Time	20.71	7.51	3.48	1.62
FF—Foraging Time	78.91	67.45	52.11	33.5

Table 8. Total number of targets when varying the capacity of the robots—Scenario 2.

Capacity of the Robots	1	4	8	16
LFFA—Total Number of targets	115.4	384.3	849.7	1255.5
FF—Total Number of targets	15.6	39.6	147.6	234.6

6.3. Results and Discussion

In Scenario 1, the foraging time decreases by increasing the number of robots (see Figure 5a and Table 7). By going toward 6400 robots, the foraging time tends to be stable, especially for the LFFA case. Since targets are clustered, it is enough that a robot finds a target, and then the attraction process will attract as many neighboring robots as possible. This increases the collected targets and decreases both the time spent in random walk and the foraging time. From simulations, a fascinating behavior of recruitment is observed between robots and resulted from light emitted by robots.

In Scenario 2, the number of collected targets grows by increasing the robot capacity (see Figure 5b and Table 8). Robots in LFFA benefit from the already found cluster to collect targets until its maximum capacity. This decreases the time spent in homing and returning to cluster when the capacity of the robot is 1. Robots return to a cluster that was already found by using the last headings before depositing targets. An optimized return strategy may enhance the results. Such a strategy will be deepened in the future work.

7. Conclusions

In this paper, we have proposed a swarm intelligence-based search algorithm called Lévy walk and Firefly-based algorithm (LFA), which is a hybridization of the Lévy walk and the Firefly algorithms. The LFA has been applied to Multi-Target Search and Multi-Robot Foraging. Numerical experiments have been conducted that highlight how the proposed algorithm performs better than the original Firefly one.

In the future, we intend to carry out a study about the algorithm-dependent parameters, such as α , γ , p in the Lévy walk algorithm, and α , β , and μ , S_r , D in the Firefly

algorithm, which we fixed according to the literature without a dedicated experimental study. Moreover, we will consider energy management [25]. Finally, since the low-level obstacle avoidance model that has been used can repulse robots from rich areas, we intend to study its effects and enhance it in order to maintain the current direction of search.

Author Contributions: All authors (O.Z., A.G. and H.S.) have contributed to the manuscript equally, read and agreed to the published version. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 87–100. [\[CrossRef\]](#)
3. Wang, H.; Wang, W.; Zhou, X.; Sun, H.; Zhao, J.; Yu, X.; Cui, Z. Firefly algorithm with neighborhood attraction. *Inf. Sci.* **2017**, *382–383*, 374–387. [\[CrossRef\]](#)
4. Iureva, R.A.; Maslennikov, O.S.; Komarov, I.I. Multiagent robotic systems' ambient light sensor. *Opt. Sens.* **2017**, *10231*, [\[CrossRef\]](#)
5. Zedadra, O.; Idiri, M.; Jouandeau, N.; Seridi, H.; Fortino, G. Lévy Walk-based Search Strategy: Application to Destructive Foraging. In Proceedings of the 2018 13th International Symposium on Programming and Systems (ISPS), Algiers, Algeria, 24–26 April 2018; pp. 1–6.
6. Tran, T.; Nguyen, T.T.; Nguyen, H.L. Global optimization using Lévy flights. *arXiv* **2014**, arXiv:1407.5739.
7. 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. [\[CrossRef\]](#)
8. Ali, N.; Othman, M.A.; Husain, M.N.; Misran, M.H. A review of firefly algorithm. *ARPN J. Eng. Appl. Sci.* **2014**, *9*, 1732–1736.
9. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36. [\[CrossRef\]](#)
10. Fister, I.; Yang, X.S.; Fister, D.; Fister, I., Jr. Firefly algorithm: A brief review of the expanding literature. In *Cuckoo Search and Firefly Algorithm*; Yang, X.S., Ed.; Studies in Computational Intelligence; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; Volume 516, pp. 347–360. [\[CrossRef\]](#)
11. Lenagh, W.; Dasgupta, P. Levy Distributed Search Behaviors for Mobile Target Locating and Tracking. In Proceedings of the 19th Annual Conference on Behavior Representation in Modeling and Simulation, Charleston, SC, USA, 22–25 March 2010; pp. 103–109.
12. Fioriti, V.; Fratichini, F.; Chiesa, S.; Moriconi, C. Levy Foraging in a Dynamic Environment—Extending the Levy Search. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 98. [\[CrossRef\]](#)
13. 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. [\[CrossRef\]](#)
14. Krivonosov, M.; Denisov, S.; Zaburdaev, V. Lévy robotics. *arXiv* **2016**, arXiv:1612.03997.
15. 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.
16. De Rango, F.; Palmieri, N. A swarm-based robot team coordination protocol for mine detection and unknown space discovery. In Proceedings of the 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC), Barcelona, Spain, 8–10 October 2012; pp. 703–708. [\[CrossRef\]](#)
17. Sutantyo, D.K.; Kernbach, S.; Levi, P.; Nepomnyashchikh, V.A. Multi-robot searching algorithm using Lévy flight and artificial potential field. In Proceedings of the 2010 IEEE Safety Security and Rescue Robotics, Bremen, Germany, 26–30 July 2010; pp. 1–6. [\[CrossRef\]](#)
18. Sutantyo, D.; Levi, P.; Moslinger, C.; Read, M. Collective-adaptive Lévy flight for underwater multi-robot exploration. In Proceedings of the 2013 IEEE International Conference on Mechatronics and Automation, Takamatsu, Japan, 4–7 August 2013; pp. 456–462. [\[CrossRef\]](#)

19. Fujisawa, R.; Dobata, S. Lévy walk enhances efficiency of group foraging in pheromone-communicating swarm robots. In Proceedings of the 2013 IEEE/SICE International Symposium on System Integration, Kobe, Japan, 15–17 December 2013; pp. 808–813. [\[CrossRef\]](#)
20. Yang, X.S.; Deb, S. Eagle Strategy Using Lévy Walk and Firefly Algorithms for Stochastic Optimization. In *Studies in Computational Intelligence*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 284, pp. 101–111. [\[CrossRef\]](#)
21. Hassanzadeh, T.; Kanan, H.R. Fuzzy FA: A Modified Firefly Algorithm. *Appl. Artif. Intell.* **2014**, *28*, 47–65. [\[CrossRef\]](#)
22. Yu, S.; Zhu, S.; Ma, Y.; Mao, D. A variable step size firefly algorithm for numerical optimization. *Appl. Math. Comput.* **2015**, *263*, 214–220. [\[CrossRef\]](#)
23. 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. [\[CrossRef\]](#)
24. Zedadra, O.; Guerrieri, A.; Seridi, H.; Fortino, G. A Lévy Walk and Firefly Based Multi-Robots Foraging Algorithm. In Proceedings of the International Conference on Internet and Distributed Computing Systems, Naples, Italy, 10–12 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 213–222.
25. Zedadra, O.; Seridi, H.; Jouandeau, N.; Fortino, G. An Energy-Aware Algorithm for Large Scale Foraging Systems. *Scalable Comput. Pract. Exp.* **2016**, *16*, 449–466. [\[CrossRef\]](#)