



# Lévy Walk-based Search Strategy: Application to Destructive Foraging

Zedadra Ouarda, Meysa Idiri, Nicolas Jouandeau, Hamid Seridi, Fortino Giancarlo

## ► To cite this version:

Zedadra Ouarda, Meysa Idiri, Nicolas Jouandeau, Hamid Seridi, Fortino Giancarlo. Lévy Walk-based Search Strategy: Application to Destructive Foraging. IEEE International Symposium on Programming and Systems, Jul 2018, Alger, Algeria. hal-02317295

**HAL Id: hal-02317295**

**<https://hal.science/hal-02317295>**

Submitted on 17 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Lévy Walk-based Search Strategy: Application to Destructive Foraging

Ouarda Zedadra  
LabSTIC, 8 may 1945 University  
Computer science Department,  
P.O.Box 401, 24000 Guelma, Algeria  
Email: zedadra\_nawell@yahoo.fr

Hamid Seridi  
LabSTIC, 8 may 1945 University  
P.O.Box 401, 24000 Guelma, Algeria  
Email: hamidseridi@yahoo.fr

Meysa Idiri  
Computer science Department,  
8 may 1945 University,  
P.O.Box 401, 24000 Guelma, Algeria  
Email: idirimeysa@gmail.com

Fortino Giancarlo  
DIMES, Università della Calabria  
Via P. Bucci, cubo 41c-87036. Rende, Italy  
Email: giancarlo.fortino@unical.it

Nicolas Jouandeau  
LIASD, Paris8 University,  
Saint Denis, France  
Email: n@ai.univ-paris8.fr

**Abstract**—In this paper a Search strategy based on Lévy Walk is proposed. Lévy Walk increases the diversity of solutions, and constitutes good strategies to move away from local to global search. The amount of exploration and exploitation and the fine balance between them determine the efficiency of search algorithm. The highly diffusive behavior of the original Lévy Walk algorithm results in more global search. Thus, in the proposed algorithm, the time spent in local search is increased according to the fluctuation of the searched region. Moreover, a case study on destructive foraging is presented in this paper, with the aim to apply it to several swarm robotics problems. In order to present simulations in a finite two dimensional landscape with a limited number of clustered and scattered targets, the ArGOS simulator has been used.

## I. INTRODUCTION

Swarm robotics tasks such as search and rescue [1], space exploration [2], foraging [3] [4] and surveillance [5], are based on efficient search [6]. As animals and insects, the swarm of robots is limited in capabilities, with no knowledge of the environment or the position of targets and no communication and cooperation. Searching for targets when there is insufficient time to explore the entire search space and limited knowledge about the search space involves necessarily random walks. Random walk can be used for both local and global search, depending on step sizes and the way to generate new solutions. The fine tune between intensification (exploitation) and diversification (exploration) is very essential for any search algorithm to perform well [7]. Too little exploration and too much exploitation could cause the system to be trapped in local optima which makes it very difficult to find all the targets over the whole search space. While exploration via randomization provides a good way to move away from local search to global search, giving the chance to search on other regions of the search space. The optimal trade-off between exploration and exploitation depends on the distribution of targets. Lévy random walks are assumed to be optimal when targets are sparsely distributed. Lévy walk is used in local and global search with a simple tuning of its cauchy scale parameter to get small steps in exploitation state. A Lévy pattern in a swarm of walkers allows the individual to visit new sites that the swarm

has not visited [8].

Lévy walk presents best random walk strategies when targets are sparsely distributed in the search space. However, due to the super diffusive behavior they present, some tuning is needed to improve their performance, specifically when targets are clustered. We attempt to minimize the super diffusive behavior of Lévy walk through increasing the time spent in local search (or intensification) whenever a target is located by using a satisfaction parameter. The algorithm presented here is the first stage towards a new Firefly-based [9] search algorithm that we are developing. The proposed algorithm is used to guarantee the global random search and to avoid the intensive search provided by Firefly algorithm. Swarm robotics are considered as Multi-Agent Systems (MAS) with real robots and their validation need robotic frameworks rather than MAS oriented framework [10]. To prove the effectiveness of the enhanced Lévy Walk algorithm regarding the original one, a set of computer simulations in the ARGoS framework [11] are defined and executed.

The remainder of the paper is organized as follows: the related works are discussed in Section II. Then, we present the proposed algorithm and the behavioral model of robots in Section III. We discuss the results in Section IV and conclude in Section V.

## II. BACKGROUND

Lévy walks are random walk patterns, whose step length is drawn from the Lévy distribution often in terms of a simple power-law formula. The direction of search at each step is drawn from a uniform distribution and is independent with respect to the previous steps [8]. They are efficient random walks in exploring unknown, large scale search space. They ensure properties such as robustness, scalability and adaptability. Two alternatives of the Lévy patterns are: Lévy flight and Lévy walk. A Lévy flight motion is a result of visiting disconnected points through jumping or flying. While a Lévy walk motion results from considering space between the disconnected points to be part of the searched area [12]. Since we use a swarm of walking robots, we consider a Lévy

walk motion in the proposed algorithm. We present in below the works which constitute the background of the proposed algorithm.

Authors in [13], optimize the Lévy Flight algorithm with a novel adaptation strategy based on the Firefly Optimization algorithm (FO) [14] in order to locate targets in underwater environments. Two modifications were done on the original algorithm. The first one concerns the attractiveness parameter. The second one consists in the replacement of the random motion of the FO with a random number from the Lévy distribution. The authors provide simulation-based experiments and real experiments with a limited number of robots.

Katada et al. consider in [15] a multi-target detection problem. They employ a random walk and Lévy Flight in a subsumption architecture [16]. Robots communicate with a base station through relay robots due to the multi-hop transmission of wireless communication. They study the effect of the step size of the random walk and the number of robots in a target detection problem. One step in the random walk strategy is set to 6 seconds. A step size in Lévy Flight is determined according to a Lévy probability distribution where the maximum step size is set to 30.

Dimidov et al. study in [17] Correlated Random Walk (CRW) and Lévy Walk (LW). They address a hybrid form that joins together the non-uniform distribution of tuning angles and the heavy-tail distribution of step lengths. So that both correlated movements and long relocations can be obtained at the same time.

### III. PROPOSED ALGORITHM

In the proposed search algorithm, we consider two states: *exploration* and *exploitation*. In both states, step lengths are generated by *cauchy distribution* given by Equation 1. However, the *cauchy scale* is fixed to 0 and 10 in exploration and exploitation states respectively to generate large and small step lengths. Then, the novelty we made to the original Lévy walk algorithm, is to increase the time spent in intensification each time a target is found, through a satisfaction parameter in robots. Robots are limited in computation and storage capabilities.

The proposed algorithm can be summarized in the following steps. A pseudo code of the detailed algorithm is presented in Algorithm 1 and the State Diagram of the robot's controller is shown in Figure 1.

- 1) Step1: walk straight for a step length generated by *cauchy distribution* given by Equation 1.
- 2) Step2: start an intensive search for time  $T$  and whenever  $Satisfaction \neq 0$ , while increasing the satisfaction each time a target is found and decreasing it when no target is found. When  $T = 0$  and  $Satisfaction = 0$ , return to Step 1, else repeat Step 2.

$$Q(p; x, \gamma) = x + \gamma * \tan[\pi * (p - 1/2)] \quad (1)$$

where :

–  $x$  is *cauchy location (real)*;

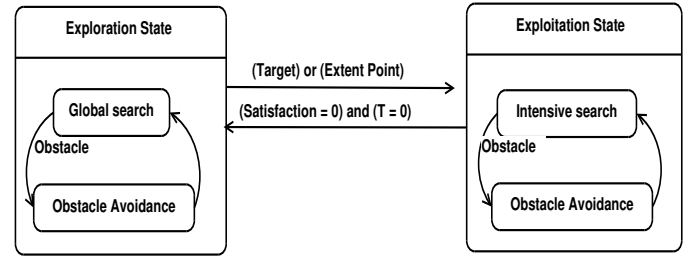


Fig. 1. State diagram of the controller

- $\gamma > 0$  is *cauchy scale (real)*;
- $p \in (-\infty, +\infty)$ ;

---

#### Algorithm 1: Pseudo code of the proposed Lévy-based Search Algorithm

---

**Data:**  $R_n, T_{max}, T_n = 0$   
**Result:**  $T_n$

- 1 Initialize population with random positions and constant velocities;
- 2 **Exploration State**
- 3 **repeat**
- 4   Generate a new step length and move randomly using Equation 1;
- 5 **until**  $(\exists \text{ Obstacle}) \parallel (\exists \text{ Target}) \parallel (\text{Extent point})$ ;
- 6 **if**  $(\exists \text{ Obstacle})$  **then**
- 7   Goto 25;
- 8 **else**
- 9   Goto 10;
- 10 **Exploitation State**
- 11  $Satisfaction \leftarrow 0; T \leftarrow T_{max}$ ;
- 12 **repeat**
- 13   **if**  $(\exists \text{ Target})$  **then**
- 14      $Satisfaction \leftarrow Satisfaction + 1$ ;
- 15      $T_n \leftarrow T_n + 1$ ;
- 16     Destruct target;
- 17   **else**
- 18      $Satisfaction \leftarrow Satisfaction - 1$ ;
- 19   **if**  $(\exists \text{ Obstacle})$  **then**
- 20     Goto 25;
- 21   **else**
- 22     Generate a new step length and move randomly using Equation 1;
- 23      $T \leftarrow T - 1$ ;
- 24 **until**  $(Satisfaction = 0) \text{ and } (T = 0)$ ;
- 25 Goto 2;
- 26 **Obstacle Avoidance State**
- 27 Turn with fixed degree;
- 28 Return to the related state (Goto 10: or Goto 2:)

---

### IV. SIMULATION-BASED EXPERIMENTS

In this study, we focus on the search of multiple targets (uniformly distributed and clustered) by a swarm of  $N$  robots. The experimental simulation is implemented in the multi-physics robot simulator ARGoS [11] which is able to simulate efficiently large-scale swarms of robots of any kind. We

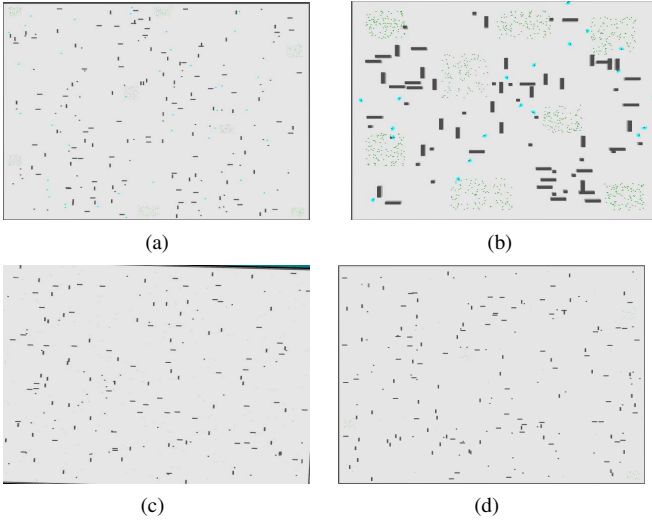


Fig. 2. Search space of the three scenarios in Table I

perform several computer simulations to test the performance of the proposed algorithm varying several criteria (number of robots, distributions of targets and environment size). The performances of the *proposed algorithm* are compared with the performances obtained by *Lévy walk* algorithm. As these are stochastic algorithms, data presented here are representative of 30 simulation executions where the STandard Deviation (STD) is reported to ensure the probabilistic quality of results.

Robots, environment and targets are modeled as follows:

- Robots: are footbots represented by a circle of radius  $r_R = 8cm$ , a higher of  $H_R = 15cm$ , Proximity Sensors  $PS_R = 12$  frontal sensor, Proximity Sensor Ring Range  $ProximityRange_R = 0.3f$ . Robots can sense targets and obstacles in their range of vision. They can move with constant velocity  $v = 80cm/s$ . They do not communicate with each other. They avoid other robots and obstacles when encountered by changing the angle of motion. Initial positions of robots are uniformly distributed in the search space.
- Environment: is a finite two dimensional space. It contains obstacles in rectangular and square forms, distributed uniformly in the search space.
- Target: represented by a circle of radius  $r_t = 0.05f$  with green color. Targets are uniformly distributed or grouped in clusters with a predefined number of targets.

#### A. Experimental Setups

Table I illustrates the realized simulation scenarios. Simulations are run for 480 seconds in the three scenarios. In scenarios 1 and 2, the number of targets in each cluster is 100 targets. In scenario 3, we perform two sub-scenarios: in the first we uniformly distribute targets and we vary them from 40 to 200 target, in the second, we group the 200 target in clusters and we vary the number of clusters from 2 – 10. Each simulation is run for 480s and the obstacle density in all simulations is 8%. The simulation search space is shown in Figure 2.

TABLE I. PARAMETERS OF SCENARIO 1, SCENARIO 2, SCENARIO 3 AND SCENARIO 4

| Parameter                                  | Value                    |
|--|--------------------------|
| <b>Scenario 1: Number of Found Targets</b> |                          |
| Robots Number                              | 20, 40, 60, 80, 100      |
| Target Distribution                        | clustered                |
| Targets Number                             | 800                      |
| Clusters Number                            | 8                        |
| Environment size                           | 120m × 120m              |
| <b>Scenario 2: Number of Found Targets</b> |                          |
| Robots Number                              | 20                       |
| Target Distribution                        | clustered                |
| Targets Number                             | 800                      |
| Clusters Number                            | 8                        |
| Environment size                           | 20m × 20m to 240m × 240m |
| <b>Scenario 3: Targets Distribution</b>    |                          |
| Robots Number                              | 30                       |
| Target Distribution                        | clustered                |
| Targets Number                             | 200                      |
| Clusters Number                            | 2 – 10                   |
| Environment size                           | 120m × 120m              |
| <b>Scenario 4: clustered vs Uniform</b>    |                          |
| Robots Number                              | 30                       |
| Target Distribution                        | Uniform – clustered      |
| Targets Number                             | 40, 80, 120, 160, 200    |
| Clusters Number                            | 2 – 10                   |
| Environment size                           | 120m × 120m              |

TABLE II. NUMBER OF FOUND TARGETS ANALYSIS (SCENARIO 1)

|           | 20     | 40     | 60    | 80    | 100    |
|-----------|--------|--------|-------|-------|--------|
| Proposed  | 341,5  | 434,97 | 591,3 | 704,2 | 783,16 |
| STD       | 8,09   | 6,06   | 5,97  | 6,48  | 4,86   |
| Lévy walk | 172,97 | 275,47 | 482,7 | 579,6 | 694,57 |
| STD       | 15,92  | 15,66  | 15,92 | 23,56 | 26,39  |

#### B. Results and Discussion

- 1) The total number of found targets increases while increasing the robots number (see Table II and Figure 3(a)). With the parameters that have been considered in scenario 1 (see Table I) over 100 robots, the total number of found targets tends to be stable. The proposed algorithm overcomes the original one at least by 80 target and at maximum by 168 targets. We reported here the STD to measure the dispersion of simulations regarding the average one. In the proposed algorithm, the STD is  $< 10$ , the results are very close to the average, thus they are significant. In Lévy walk algorithm simulation, the variation is normal (STD between 10 – 20), but it is important in the two last simulations (STD between 20 – 30).
- 2) The total number of found targets decreases while the environment size increases (see Table III and Figure 3(b)). The proposed algorithm overcomes the original one with 52 to 234 targets. Results stabilize when the size of environment is over  $240m \times 240m$ . The variations in STD go till normal in the proposed algorithm, while they are normal in the original one.
- 3) When increasing the number of clusters, the total number of found targets slowly increases (see Table IV and Figure 3(c)). In the proposed algorithm,

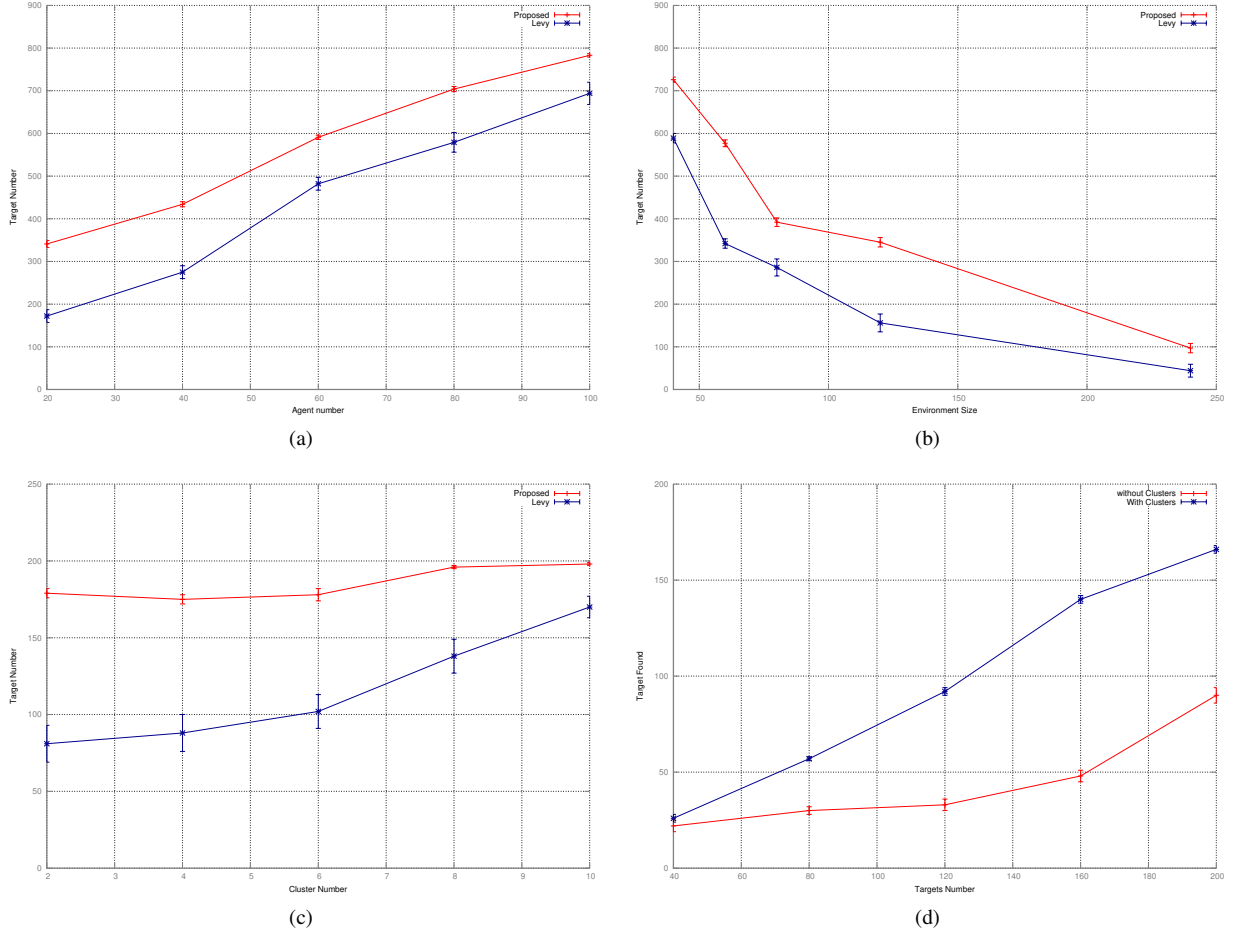


Fig. 3. Simulation results : (a) Results of scenario 1, (b) Results of scenario 2, (c) Results of scenario 3, (d) Results of scenario 4.

TABLE III. NUMBER OF FOUND TARGETS ANALYSIS (SCENARIO 2)

|                 | 40 × 40 | 60 × 60 | 80 × 80 | 120 × 120 | 240 × 240 |
|-----------------|---------|---------|---------|-----------|-----------|
| <i>Proposed</i> | 726,03  | 577,27  | 392,2   | 345       | 97,7      |
| <i>STD</i>      | 6,23    | 8,82    | 10,69   | 11,16     | 11,34     |
| Lévy walk       | 589,69  | 342,53  | 286,57  | 156,07    | 44,83     |
| <i>STD</i>      | 11,11   | 11,21   | 20,55   | 21,73     | 15,78     |

TABLE IV. CLUSTERED DISTRIBUTION OF TARGETS (SCENARIO 3)

|                 | 2      | 4      | 6      | 8      | 10     |
|-----------------|--------|--------|--------|--------|--------|
| <i>Proposed</i> | 179,06 | 175,27 | 178,77 | 196,73 | 198,47 |
| <i>STD</i>      | 3,98   | 3,93   | 4,12   | 1,96   | 1,5    |
| Lévy walk       | 81,37  | 88,47  | 102,4  | 138,97 | 170,87 |
| <i>STD</i>      | 12,8   | 12,58  | 11,95  | 11,23  | 7,753  |

the increase reaches its maximum 17 *targets* when number of clusters is increased from 6 to 8. It has decreased with 4 *targets* when the number of clusters is increased from 2 to 4. While in the original algorithm the increase is fast, it reaches 36 *target* when increasing the number of clusters from 6 to 8. The proposed algorithm overcomes the original one with a maximum difference of 97 *target* when the number of clusters is 2. Results in the proposed algorithm are very close (STD between 1–4), while they are normal in the original algorithm (STD between 7 – 12).

- 4) The total number of found targets increases while increasing the number of targets dispersed in the environment in the two configurations (see Table V and Figure 3(d)). This number is greater when targets are clustered. The difference reaches its highest value

(92 *target*) when the number of targets is 160 and the number of clusters in the second configuration is 8. When targets are clustered, results are better from the uniform distribution due to the incitation on exploitation while the region is rich of objects. Results in the two configurations are very close (STD between 1 – 4).

## V. CONCLUSION

In this paper, we enhanced the performance of the original Lévy walk algorithm in a destructive foraging scenario. The proposed Lévy walk-based strategy presents a significant improvement due to the increase in the exploitation state whenever a target is found. The results obtained by the proposed algorithm overcome the Lévy walk algorithm in the different

TABLE V. CLUSTERED VERSUS UNIFORM DISTRIBUTION OF TARGETS IN THE PROPOSED ALGORITHM (SCENARIO 3)

|                  | 2     | 4    | 6     | 8      | 10    |
|------------------|-------|------|-------|--------|-------|
| Without Clusters | 22,1  | 30,5 | 33,13 | 48,17  | 90,1  |
| STD              | 3,38  | 2,42 | 3,66  | 3,89   | 4,02  |
| With Clusters    | 26,13 | 57   | 92,93 | 140,37 | 166,2 |
| STD              | 2,29  | 1,79 | 2,42  | 2,6    | 2,91  |

presented configurations. In particular, the proposed algorithm is more efficient in a clustered target environment.

The proposed algorithm is the first step towards a new search strategy based on a bio-inspired algorithm (Firefly) which can be applied to several robotic applications. Currently, we are working on the interaction protocol to allow cooperation between robots. We intend in the future to: (1) consider the limitation of robot's energy as in our previous works [18], (2) adapt and test the efficiency of the proposed algorithm on foraging scenarios [19].

## REFERENCES

- [1] M. Senanayake, I. Senthooan, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed, "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems*, vol. 75, pp. 422–434, 2016.
- [2] O. Zedadra, H. Seridi, N. Jouandeau, and G. Fortino, "A distributed foraging algorithm based on artificial potential field," in *2015 12th International Symposium on Programming and Systems (ISPS)*. IEEE, apr 2015, pp. 1–6.
- [3] A. F. Winfield, *Foraging Robots*. New York, NY: Springer New York, 2009, vol. 6, pp. 3682–3700.
- [4] O. Zedadra, H. Seridi, N. Jouandeau, and G. Fortino, "A Cooperative Switching Algorithm for Multi-Agent Foraging," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 302–319, apr 2016.
- [5] A. Kolling and S. Carpin, "Multi-robot surveillance: An improved algorithm for the GRAPH-CLEAR problem," in *2008 IEEE International Conference on Robotics and Automation*, no. 978. IEEE, may 2008, pp. 2360–2365.
- [6] O. Zedadra, N. Jouandeau, H. Seridi, and G. Fortino, "Stigmergic MASA: A Stigmergy Based Algorithm for Multi-Target Search," in *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems*, vol. 2, sep 2014, pp. 1477–1485.
- [7] V. Méndez, D. Campos, and F. Bartumeus, *Stochastic foundations in movement ecology: anomalous diffusion, front propagation and random searches*. Springer Science & Business Media, 2013.
- [8] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, "Lévy flight search patterns of wandering albatrosses," *Nature*, vol. 381, no. 6581, pp. 413–415, may 1996.
- [9] X. S. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, no. 1, pp. 36–50, 2013.
- [10] G. Fortino, A. Garro, and W. Russo, "An integrated approach for the development and validation of multi-agent systems," *Computer Systems Science & Engineering (CSSE)*, vol. 20, no. 4, pp. 259–280, 2005.
- [11] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, dec 2012.
- [12] M. F. Shlesinger and J. Klafter, "Lévy walks versus lévy flights," in *On growth and form*. Springer, 1986, pp. 279–283.
- [13] D. Sutanty, P. Levi, C. Moslinger, and M. Read, "Collective-adaptive Lévy flight for underwater multi-robot exploration," in *2013 IEEE International Conference on Mechatronics and Automation*. IEEE, aug 2013, pp. 456–462.
- [14] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5792 LNCS, pp. 169–178.
- [15] Y. Katada, A. Nishiguchi, K. Moriwaki, and R. Watakabe, "Swarm robotic network using Lévy flight in target detection problem," *Artificial Life and Robotics*, vol. 21, no. 3, pp. 295–301, sep 2016.
- [16] R. Brooks, "A robust layered control system for a mobile robot," *IEEE journal on robotics and automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [17] C. Dimidov, G. Oriolo, and V. T. B., *Swarm Intelligence*, ser. Lecture Notes in Computer Science, M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle, Eds. Cham: Springer International Publishing, 2016, vol. 9882.
- [18] O. Zedadra, H. Seridi, N. Jouandeau, and G. Fortino, "An Energy-Aware Algorithm for Large Scale Foraging Systems," *Scalable Computing: Practice and Experience*, vol. 16, no. 4, pp. 449–466, jan 2016.
- [19] O. Zedadra, N. Jouandeau, H. Seridi, and G. Fortino, "Multi-Agent Foraging: state-of-the-art and research challenges," *Complex Adaptive Systems Modeling*, vol. 5, no. 1, pp. 1–24, dec 2017.