Jurnal Teknologi

Full Paper

COOPERATIVE SOURCE DETECTION USING AN OPTIMIZED DISTRIBUTED LEVY FLIGHT ALGORITHM

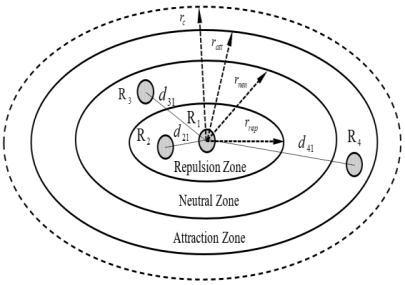
Mad Helmi Ab. Majid[a]*, Mohd Rizal Arshad[b], Mohd Faid Yahya[c]

[a]Faculty of Computer and Meta-Technology, UPSI, 35900, Tanjong Malim, Perak, Malaysia
[b]School of Electrical and Electronic Engineering, Engineering Campus, USM, 14300, Nibong Tebal, Penang, Malaysia
[c]Faculty of Electrical Engineering, UTEM, 76100, Melaka, Malaysia

Article history
Received
10 June 2022
Received in revised form
30 July 2023
Accepted
6 September 2023
Published Online
20 December 2023

*Corresponding author
madhelmi@meta.upsi.edu.my

Graphical abstract



Abstract

Source signal detection plays important roles in many real-world target searching problems. Source detection is necessary before a full search process utilizing the detected signal can be performed where minimizing the detection time and maximizing the search space exploration or coverage are the main problems. In this paper, an optimized Levy Flight algorithm known as a Distributed Levy Flight (DLF) for swarm agents is proposed. The DLF algorithm is optimized by means of repulsive artificial potential force to disperse the agents in order to optimize the search space coverage and detection time. Additionally, to integrate cooperative behavior, an artificial attractive force is used to maintain communication among the agents. The results showed that the proposed DLF algorithm successfully improve detection time (113.1s) and area coverage (78.3%) compared to the existing algorithms: Brownian Walk (325.5s, 31.7%), Correlated Random Walk (356.2s, 35.1%), Levy Flight (201.3s, 56.6%), Levy Flight with Artificial Potential Fields (151.9s, 70.2%).

Keywords: Levy flight, source detection, source search, swarm robots, random search

Abstrak

Pengesanan isyarat sumber memainkan peranan penting dalam masalah pencarian sasaran dalam dunia sebenar. Pengesanan sumber diperlukan sebelum proses pencarian penuh menggunakan isyarat yang dikesan boleh dilakukan di mana meminimumkan masa pengesanan dan memaksimumkan penerokaan atau liputan ruang carian adalah masalah utama. Dalam kertas kerja ini, algoritma Penerbangan Levi yang optimum yang dikenali sebagai Penerbangan Levi Teragih (DLF) untuk ejen kawanan dicadangkan. Algoritma DLF dioptimumkan melalui daya potensi buatan untuk menyebarkan agen untuk mengoptimumkan liputan ruang carian dan masa pengesanan. Selain itu, untuk mengintegrasikan perilaku koperatif, daya tarikan buatan digunakan untuk mengekalkan komunikasi antara ejen. Keputusan menunjukkan bahawa algoritma DLF yang dicadangkan berjaya mengurangkan masa pengesanan (113.1s) dan liputan kawasan (78.3%) berbanding dengan algoritma sedia ada: Brownian Walk (325.5s, 31.7%), Correlated Random Walk (356.2s, 35.1%), Penerbangan Levi (201.3s, 56.6%), Penerbangan Levi dengan Medan Potensi Buatan (151.9s, 70.2%).

Kata kunci: Penerbangan Levy, pengesanan sumber, pencarian sumber, robot kawanan, pencarian rawak

1.0 INTRODUCTION

The research of swarm robotics in many aspects of tasks has shown significant gain of momentum in recent years due to the large potential of future applications. Moreover, the scalability, flexibility and robust characteristics of swarm robotics emerges as a result of simple control and coordination rules that make it ideal for solving complex tasks and cutting-edge applications [1,2]. One of the tasks is target searching which is important in many applications such as search and rescue operation, landmines cleaning and chemical or radiation leakage detection [3,4]. In an underwater environment, searching for a flight black box after a flight crash, retrieving a wrecked underwater vehicle and tracking of marine life are the examples of target search [5]. These examples of source searching applications demonstrate the complexity of the operation.

Naturally, any type of source has limited detectable intensity due to limited sensitivity of the used sensor. In other words, the source signal can only be detected if the intensity is strong enough. Typically, target search involves two steps of operations known as exploration and exploitation. In the exploration step, robots are required to explore the area in order to detect the target's signal. In this case, the used algorithms must be independent of signal intensity and the execution of the exploration algorithm is terminated once the signal is successfully detected by the swarm. Exploitation on the other hand used the detected signal and iteratively moved closer towards the source. The exploitation step can only take place once the target signal is successfully detected. Thus, the actual time taken to search and locate a target include both exploration time and exploitation time. In both cases, optimizing the time taken always become priority as the fact that all robotics systems have limited power supply and the aims of the searching process itself is critically important such as to save lives (e.g., search and rescue after earthquake) or valuable items (i.e., search and recovery of underwater vehicles). As a result, algorithms that optimize the time taken to complete both exploration and exploitation are always needed.

There are many exploitation methods that have been widely studied and proposed based on swarm intelligence (SI) such as Particle Swarm Optimization [6-8], Ant Colony Optimization [9], Bean Optimization [10] and Bacteria Chemotaxis [11]. Other exploitation methods include pattern formation [12], infotaxis [13], behavior-based [14] and hybrid method [15]. These algorithms require a continuous detectable source signal intensity and will fail to work in case of the source signal is not detected.

However, less attention has been given to the exploration step. Most researchers assumed that when the source signal is not detected, robots just perform random movements without considering the optimality of the algorithm. Thus, the actual time spend to locate the source is not optimize since random based algorithms such as random walk (RW) or Brownian walk [16], correlated random walk (CRW), biased random walk (BRW) [17], and Lévy flight (LF) [18] naturally have no embedded mechanism to ensure optimization of the time taken. However, among these algorithms, LF has been proven to have better area exploration capability compared to RW or any other random-based algorithms. This is due to the fact that LF has long jump characteristics which increase the possibility of visiting an unexplored area as observed in nature such as when animals search for sparsely located food [19]. The long jump characteristics are characterized by a power law distribution that controls the frequency of long jumps generation. In [20], LF is proven as a successful detection algorithm but the optimality of the algorithm is not studied. Khaluf *et al.* used LF as a collective detection algorithm but the optimality and communication assurance are not thoroughly studied [21].

In general, LF has several disadvantages that need to be overcomed so that it can be implemented as an effective and optimal cooperative detection algorithm, has a good area coverage capability and provides optimal target signal detection. Firstly, at the individual level, LF has characteristics that can possibly cause robots to immediately revisit the previously visited area and has a tendency to be stuck at the boundary of search space when the next waypoint is located outside of the search space. As a result, the area exploration is not optimized and the time taken to detect the source will increase. However, these problems have been addressed in our previous work by introducing angle constraint and boundary reflection mechanism [22].

Similarly, at the swarm level, multiple robots revisiting the same area will reduce the detection efficiency and thus, overall searching time. To overcome this problem, each robot should have some state information of other robots in order to avoid revisiting the previously explored area. Additionally, LF by its nature is not a cooperative algorithm because it defines individual movement instead of collective movement and its execution is independent of requiring information from other agents [23]. To overcome these problems, a mechanism to disperse the robots in the search space is needed and a communication network among the robots will be established in this study and maintained throughout the searching process. This will allow agents to communicate, share detection information and consequently, make decisions to optimize the source's signal detection process. In this research work, a Distributed Lévy Flight (DLF) is proposed to solve the mentioned problems. The concept of repulsive artificial potential force is used to disperse the robots in the search space such that robots can minimize the cooccurrences of revisiting

the similar areas multiple times. Additionally, attractive artificial potential force is used to maintain communication among the agents throughout the searching process. The combination of both forces will ensure robots in the swarm will be able to maximize exploration of unexplored areas and at the same time maintain their communication for sustaining cooperative behavior. Additionally, the optimal values of DLF parameters were also determined in this work based on performance measurement parameters.

To evaluate performance of the proposed algorithm, we consider an application of underwater acoustic source detection and Autonomous Surface Vehicles (ASVs) as a swarm robotics platform. The experiments were conducted through simulations where ASV is represented by a mathematical model. The rest of this paper is organized as follows: methodology, results and discussion and conclusions.

2.0 METHODOLOGY

The detection problem is illustrated in Figure 1. In this study, the robot platform is ASV and the interested target to be detected is an underwater acoustic source. Each robot or agent is equipped with an acoustic detection sensor (i.e., hydrophone), an RF-based wireless communication module and a Global Positioning System (GPS) as a position sensor. Like many other types of sources, the detection range of underwater acoustic sources is limited by source signal strength and sensitivity of the sensor used to detect the signal. In this case, the acoustic source signal can only be detected by hydrophone if robots successfully move within the detectable radius of the source.

Once deployed in a bounded search space, each ASV will follow the waypoints generated by DLF to explore the search space and at the same time sample the source signal. Throughout this process, ASVs will communicate with each other to share and exchange detection information. Thus, the communication network should be maintained throughout the search process to ensure the success of cooperation among the ASVs.

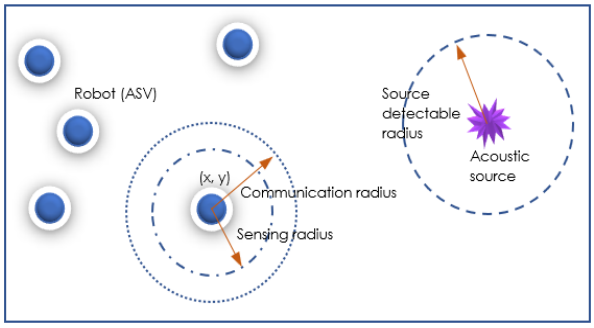


Figure 1 Source detection problem in a bounded search space

In order to detect the source, each agent (in our case ASV) has to explore the search environment by following the generated waypoints. For this purpose, the source detection algorithm (SDA) based on Distributed Levy Flight (DLF) is proposed to guide the ASV to navigate within the search environment. The waypoints are generated by DLF where the next waypoint, $p_k = (x_k, y_k)$ at the k^{th} update step can be calculated by

$$\begin{aligned} x_k &= x_{k-1} + l_k \cos(\varphi_k) \\ y_k &= y_{k-1} + l_k \sin(\varphi_k) \end{aligned} \tag{1}$$

where l_k is the step length, $x_{min} \le x_k \le x_{max}$ and $y_{min} \le y_k \le y_{max}$ where $[x_{min}, x_{max}]$ and $[y_{min}, y_{max}]$ define the boundary of a search space. The turning angle, φ_k is sampled directly from a continuous uniform distribution, U. The probability of generating long jumps or steps length, l_k in LF is determined by the characteristic of a power law distribution given by

$$P(l) = C\, l^{-\mu} \quad \text{for } l \ge l_{\min} \text{ and } 1 < \mu \le 3 \tag{2}$$

where l_{min} is the minimum flight length limit and μ is the power law exponent. C is known as a normalization constant and it can be expressed as

$$C = (\mu - 1)\, l_{\min}^{\mu - 1} \tag{3}$$

From (2) and (3), we obtain

$$P(l) = \frac{\mu - 1}{l_{\min}} \left(\frac{l}{l_{\min}} \right)^{-\mu} \quad \text{for } l \ge l_{\min} \tag{4}$$

The parameter l_{min} is introduced in the DLF model in order to consider the limits and physical capability of the robots and the size workspace under consideration. For example, without this limit, the generated step length of 0.1 m is difficult to be followed by a large size robot platform and it is not practical for generating waypoints in a large search space such as lake, ocean, etc.

To generate probability of step length in (4), it must be converted into commonly used probability, compatible and available in simulation software such as a normal distribution, U. If the cumulative distribution function (CDF) of the desired distribution is represented by F_{CDF}, then

$$l = F_{CDF}^{-1}(U) \tag{5}$$

The CDF of a power-law distribution is given by

$$F_{CDF}(l) = \int_{l_{\min}}^{\infty} C_n\, l^{-\mu} = 1 - \left(\frac{l}{l_{\min}} \right)^{1-\mu} \tag{6}$$

Solving for the inverse of the distribution function yields the step length in term of normal distribution given by

$$l = F_{CDF}^{-1}(U) = l_{\min}(1-U)^{\frac{1}{-\mu+1}} \quad (7)$$

From (7), there two parameters that affect the generated step length are l_{min} and μ.

Based on our previous discussion, DLF should overcome two problems of the conventional Lévy flight: avoid revisiting similar areas frequently and enable communication among the robots in the swarm. To solve these problems, DLF is defined by three different zones of robot's interaction: repulsion zone (robots will move away from each other), neutral zone (nor repulsion or attraction) and attraction zone (robot will be attracted to each other) as shown in Figure 2.

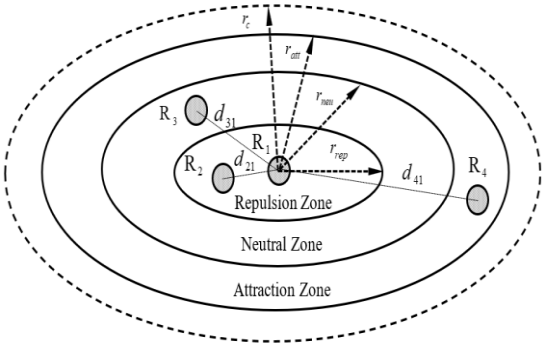


Figure 2 Repulsion, neutral and attraction zones of DLF

The zones are defined as follow:

- *Repulsion zone*: If robots are located too close to each other, they should move away from each other by iteratively repulsing from each other.
- *Attraction zone*: If a robot moves away from each other, it should be attracted towards the swarm to avoid loss of communication with at least one of the robots from the swarm to ensure the success of cooperative behavior implementation.
- *Neutral zone*: When robots are located sufficiently far away from each other and the communication signal is sufficiently strong to maintain the communication network, they should be given freedom to explore the search space without any restrictions.

The artificial repulsion and attraction forces to be used in DLF are defined by

$$F_{ij} = \begin{cases} F_r = -\dfrac{F_{\max}}{r_{rep}} d_{ij} + F_{\max} & \text{if} \quad 0 < d_{ij} \le r_{rep} \\ 0 & \text{if} \quad r_{rep} < d_{ij} \le r_{neu} \\ F_a = \dfrac{F_{\max}}{r_{att} - r_{neu}} d_{ij} + F_{\max}\left(1 - \dfrac{r_{att}}{r_{att} - r_{neu}}\right) & \text{if} \quad r_{neu} < d_{ij} \le r_{att} \end{cases} \quad (8)$$

where d_{ij} is the distance between robot i and j, F_{max} is the maximum magnitude of the interaction force, r_{rep} is the repulsive radius, r_{neu} is the neutral radius and r_{att} is the attractive radius (i.e., $r_{att} < r_c$ where r_c is the maximum communication radius). Each robot can only communicate with other robots to share information within its communication radius.

Equation (8) can be expressed in a notation vector format,

$$\boldsymbol{F}_{ij} = F_{ij}\hat{\boldsymbol{d}}_{ij} = \begin{cases} \boldsymbol{F}_{\boldsymbol{r},ij} = -F_r\hat{\boldsymbol{d}}_{ij} & \text{if} \quad 0 \le d_{ij} \le r_{rep} \\ 0 & \text{if} \quad r_{rep} < d_{ij} \le r_{neu} \\ \boldsymbol{F}_{\boldsymbol{a},ij} = F_a\hat{\boldsymbol{d}}_{ij} & \text{if} \quad r_{neu} < d_{ij} \le r_{att} \end{cases} \quad (9)$$

where $\boldsymbol{d_{ij}} = \boldsymbol{p_j} - \boldsymbol{p_i}$ and $\boldsymbol{p_j}$ and $\boldsymbol{p_i}$ is the position of robot i and robot j, respectively. For three or more robots, the resultant force experience by robot i can be expressed as a summation of interaction forces of the neighboring robots,

$$\boldsymbol{F}_i = \sum_{j=1, j\ne i}^{N_s} \boldsymbol{F}_{ij} = \sum_{j=1, j\ne i}^{N_s} \boldsymbol{F}_{\boldsymbol{r},ij} + \boldsymbol{F}_{\boldsymbol{a},ij} \quad (10)$$

The direction of the net interaction force can be computed as

$$\phi_{new,i} = \operatorname{atan2}\left(\frac{\boldsymbol{F}_i^{\,y}}{\boldsymbol{F}_i^{\,x}}\right) \quad (11)$$

in which a new or a redirected waypoint, $\boldsymbol{p_{k+1,new}}$ is generated as follow

$$\boldsymbol{p}_{k+1,new} = \boldsymbol{p}(t) + d_{adj}\begin{bmatrix}\cos(\phi_{new,i})\\ \sin(\phi_{new,i})\end{bmatrix} \quad (12)$$

where d_{adj} is the adjustment distance which is equal to the last computed generated step length.

The implementation of the DLF on the individual robot is given by Algorithm 1. From the algorithm, some parts of the algorithm which involve the improvement of the LF at the individual level were proposed and discussed in our previous work (refer to [22]). In this algorithm, equations (1) through (12) are used to execute DLF based SDA. As stated earlier, target search involves two steps of operations known as exploration and exploitation. SDA is the exploration algorithm to optimize exploration of an area while the source tracing algorithm (STA) is not included as part of this paper and will be our future work where STA can be any target signal exploitation algorithm such as Particle Swarm Optimization.

Algorithm: DLF SDA for robot *n*

Set: l_{min}, ϕ_{min}, ϕ_{max}, Θ_{min}, r_{rep}, r_{neu}, r_{att}, r_c, P_{DTH}
$\boldsymbol{p}_i(t = 0) = initialization()$, i
while (Stopping criteria is not satisfied) **do**
$[N_s, ID_s] = find\_neighbors()$
generate $I(k)$
generate $\boldsymbol{\phi}(k)$
calculate $\boldsymbol{p}(k+1)$
for $j = 1 : N_s$
  calculate $d_{ij} = ||\boldsymbol{p}_j(t) - \boldsymbol{p}_i(k)||$
  if $0 \leq d_{ij} \leq r_{rep}$
    calculate $\boldsymbol{F_{ij}} = -F_{ij} \cdot \hat{\boldsymbol{d}}_{ij} = -F_{ij} \cdot \boldsymbol{d_{ij}} / ||\boldsymbol{d_{ij}}||$
  else
    set $\boldsymbol{F_{ij}} = 0$
  end
end for
calculate $\boldsymbol{F_i}$
if $\boldsymbol{F_i} \neq 0$
 calculate $\phi_{new,i}(k+1)$
 calculate $\boldsymbol{p}(k+1)$
end if
calculate $\Theta(k)$
if $\Theta(k) < \Theta_{min}$
calculate $\phi_{new}(k)$
calculate $\boldsymbol{p}(k+1)$
end if
if $\boldsymbol{p}(k) \in \boldsymbol{p}_{max/min}$
  calculate $\hat{\boldsymbol{d}}_{k,k+1}$
  calculate $\boldsymbol{p}(k+1)$
end
$\boldsymbol{p}(k+1) = check\_boundary\_limit()$
while $(p(t) != \boldsymbol{p}(k+1))$ **do**
  calculate $d_{ij} = ||\boldsymbol{p}_j(t) - \boldsymbol{p}_i(t)||$
 if $r_{neu} < d_j \leq r_{att}$
  calculate $\boldsymbol{F_{ij}} = F_{ij} \cdot \hat{\boldsymbol{d}}_{ij}$
  calculate $\phi_{new}(k)$
  calculate $\boldsymbol{p}(k+1)$
 end if
$k = k+1$
end while
end while

As mentioned earlier, the test of the proposed algorithm is conducted by using ASVs as the swarm platforms to detect the presence of underwater acoustic sources. The idea is to replicate the process of searching a plane black box when it crashed into the sea or searching or recovery for a failure underwater vehicle. Since this study is conducted based on simulation to evaluate the proposed algorithm, a suitable model of ASV is needed. We assume that the ASV used for this purpose is a non-conventional model but a specialized design ASV for swarming application as we proposed in [24]. The illustration of the proposed ASV model is shown in Figure 3. The proposed ASV platform is equipped with a GPS module for positioning, a compass for heading, an RF module for communication and a hydrophone for detection of acoustic intensity. In the exploration process, SDA is used to generate the required waypoints to be followed by the ASV and the navigation of the ASV (represented by a mathematical model) while following the way points is guided by the sliding mode controller.

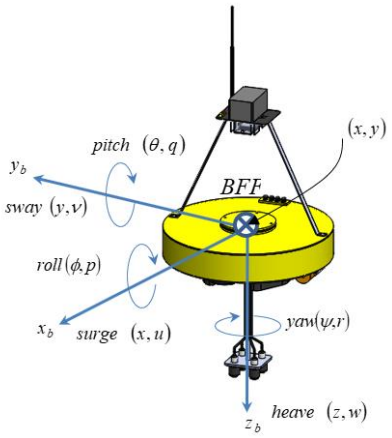


Figure 3 Physical model of the ASV for swarming application

To represent the actual ASV as a swarm platform in a virtual simulation environment, a complete mathematical model of the ASV should be derived. The dynamics model of the ASV that closely represent the physical model is given by

$$\begin{cases} \dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{v} \\ \boldsymbol{M}\dot{\boldsymbol{v}} + \boldsymbol{D}\boldsymbol{v} = \boldsymbol{\tau} \end{cases} \tag{13}$$

where $\boldsymbol{R}(\psi)$ is the rotation matrix, $\boldsymbol{v}$ is the linear-angular velocities vector, $\boldsymbol{\tau}$ is the vector of control inputs. $\boldsymbol{M}$ is the mass-inertia matrix and $\boldsymbol{D}$ is the damping matrix which are given by

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 \\ 0 & 0 & J_z - N_{\dot{r}} \end{bmatrix} \tag{14}$$

$$\boldsymbol{D} = -\begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix} \tag{15}$$

The thruster for and moment matrix is given by

$$\boldsymbol{\tau} = [\tau_u \ 0 \ \tau_r]^T \tag{16}$$

where

$$\tau_u = F_{TL} + F_{TR} \tag{17}$$

$$\tau_r = \frac{1}{2} b (F_{TL} - F_{TR}) \tag{18}$$

F_{TL} and F_{TR} is the thrust force produced by the left and right thrusters, respectively. The value for each parameter used in simulation is listed in Table 1. Based on the given model, the performance evaluation of the algorithm can be expressed in time domain instead of number of steps.

Table 1 ASV model parameters

Category	Parameters	Final Values
Mass-Inertia	J_z	0.1094
Added-mass	$X_{\dot{u}}$	-0.2151
	$Y_{\dot{v}}$	-0.1757
	$N_{\dot{r}}$	-0.0082
Damping	X_u	-0.9031
	Y_v	-0.5119
	N_r	-0.0143

The controller is designed based on methodology proposed in [25]. The used decoupled controller consists of PI speed controller given by

$$\tau_u = -K_p(u-u_d) - K_i\int_0^t (u-u_d)d\tau, \quad K_p, K_i > 0 \qquad (19)$$

where u_d is the desired velocity, K_p is the proportional term and K_i is the integral term and heading controller sliding mode heading controller

$$\tau_r = -\mathbf{k}_h^{\mathrm{T}}\mathbf{x}_\mathbf{h} + \left(\mathbf{h}_h^{\mathrm{T}}\mathbf{B}_\mathbf{h}\right)^{-1}\left(\mathbf{h}_h^{\mathrm{T}}\dot{\mathbf{x}}_{\mathbf{d}h} - \eta_{ch}\tanh\left(\frac{\boldsymbol{\sigma}_h(\mathbf{x}_e)}{\boldsymbol{\Phi}_h}\right)\right) \qquad (20)$$

where the unknown control parameters are η_{ch} and Φ_h. The value of each controller parameters used in this simulation are summarized in Table 2.

Table 2 Controller parameters

Subsystem	Controller	Parameter	Value
Propulsion	PI	K_p	140.61
		K_i	10.71
Heading	SMC	η_{ch}	3.50
		Φ_h	1.50
		p_{h1}	-1.50
		p_{h2}	-1.40

Simulation was done in MATLAB where each ASV is represented by a complete mathematical model, SDA is used to generate waypoints of the robot and controller is used to navigate the ASV to closely follow the generated way points.

3.0 RESULTS AND DISCUSSION

As seen from equation (8), there are two parameters that will affect the generated step length and its distribution. The first parameter is minimum step length, l_{min} which determines the shortest possible step length will be generated by DLF. The second parameter is the power law exponent, μ which determines the weighted distribution pattern of the tail of a power law distribution. These two parameters have a direct impact on the exploration capability of the individual robot. Thus, in this paper, the optimal value of these two parameters is determined by evaluating the average steps length and the area coverage of movement of a robot in a specified search area. The average step length, l_{avg} is defined as follow:

$$l_{avg} = \frac{1}{k_{run}k_{stp}}\sum_{j=1}^{k_{run}}\sum_{k=1}^{k_{stp}} l(k) \qquad (21)$$

where k_{stp} and k_{run} is the number of steps and number of runs (i.e., number of simulation repetitions) respectively. Note that k_{max} is the maximum number of steps set for simulation where a step is represented by distance between two waypoints generated by the SDA.

Figure 4 demonstrates the effect of l_{min} on the l_{avg} from where it can be observed that the average steps length, l_{avg} decreases as the value of μ decreases and the value of l_{avg} increases as the value of l_{min} increases. It also can be observed that as the value of μ gets closer to 1, the average step length increases as the motion mimics towards a ballistic motion and as the value of μ gets closer to 3, the average step length becomes smaller because movement of the agent tends to exhibit Brownian motion. Figure 5 illustrates the exploration capability of DLF for different value l_{min} and μ.

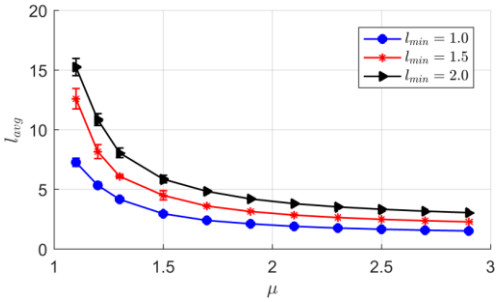


Figure 4 Average steps length for different values of l_{min} for k_{stp} = 1000 and k_{run} = 100

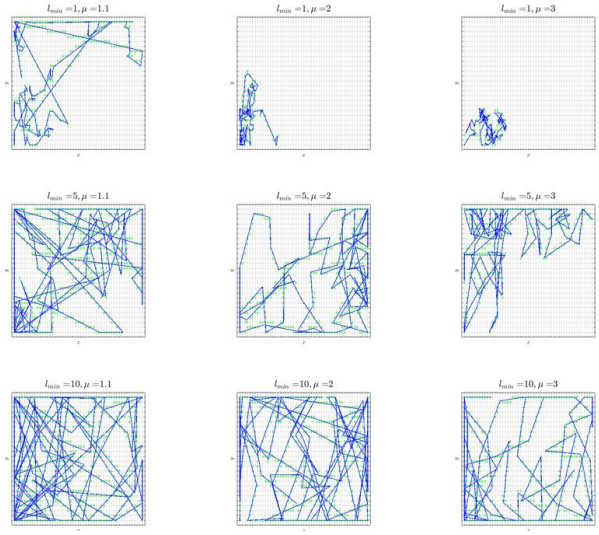


Figure 5 Exploration performance for increasing value of l_{min} from row and increasing number of μ from the column for n_A = 1 for k_{max} = 100. Blue line and green circle indicate robot trace and visited grids, respectively

From the figure, it can be observed that a small value of μ provides better exploration performance compared to a larger value of μ for a similar number of generated steps and value of l_{min}. It also can be observed that the larger the value of l_{min}, the larger the area is explored by a single agent.

Additionally, the distance travelled by an agent is longer when μ approaches 1.1 compared to μ approaches 3 regardless of the value of l_{min} as can be observed in Figure 6. However, these observations only give information of how the size of the average step length changes for different values of l_{min} and μ but the optimal value of the two parameters for optimal search space exploration cannot be fully decided yet.

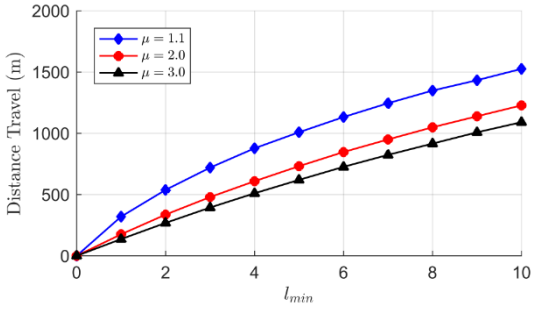


Figure 6 Distance travel by an agent for different values of l_{min} and μ in a search space $n_A = 1$ for $k_{max} = 100$

As previously mentioned, source detection can be optimized by optimizing search space exploration. The search space exploration can be optimized by minimizing the frequency of revisiting search sites that have been previously visited. Thus, to achieve optimal detection efficiency, revisiting a similar area must be minimized and the area of coverage must be maximized. Thus, the Percentage Area Coverage (PAC) and Frequency of Visiting (FoV) can be used as performance matrices to select optimal values of the LF parameters. The percentage area coverage (PAC) is calculated as a ratio of number of visited grids, NG_{vis} over a total number of grids, NG_{tot} as follow:

$$PAC = \frac{NG_{vis}}{NG_{tot}} \times 100\% \tag{22}$$

Each grid must be visited at least once to be considered as a visited grid. The result of the coverage area performance for a single robot using various values of μ is shown in Figure 7. From the figure, it can be observed that the value of PAC is initially increasing for all values of μ but after some optimal value of l_{min}, the PAC value later decreases as l_{min} increases for all values of μ. For small values of l_{min}, the value of μ corresponding to a near Ballistic motion (i.e., $\mu \rightarrow 1$) achieves better PAC performance compared to value of μ closer to the Brownian motion (i.e., $\mu \rightarrow 3$) as shown in Figure 8. This is because in a bounded search space longer steps length means robot can reach farther compared to shorter steps length. However, after some optimal values of l_{min} the PAC for all values of μ decreases as the value of l_{min} increases. The reason is that as the steps length becomes too large, the desired waypoint of the robot overpassed or frequently collided with the boundary of the search space. In this situation, robot might be temporarily stuck at the boundary while waiting for the next turning angle to turn away from the boundary.

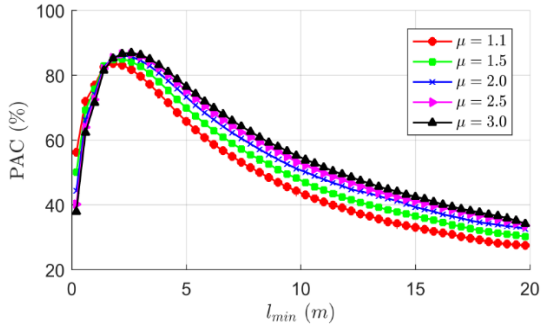


Figure 7 PAC by a single robot for different values of μ and for large values of l_{min} for $n_A = 1$

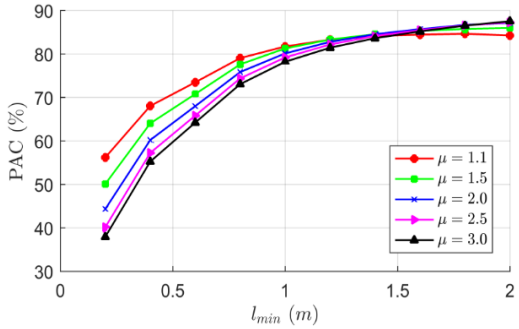


Figure 8 PAC by a single robot for different values of μ and for small values of l_{min} for $n_A = 1$ where optimal l_{min} is found to be $l_{min,opt} \approx 1.5$ m when $\mu = 1.1$

It also can be observed that for any value of l_{min} larger than the optimal value of l_{min}, a larger value of μ shows better PAC performance compared to a smaller value of μ. This is due to the fact that LF with a large value of μ tends to have smaller average steps length which has less tendency to overpass the search boundary compared to a small value of μ. Thus, for a bounded search space, a small value of μ provides better area coverage when l_{min} is small and a larger value of μ gives better area coverage when l_{min} is large.

For a specified search space, an average frequency of a visiting (FoV) is defined by a total frequency of visited grids over a total number of visited grids can be used to evaluate performance of the robot. Mathematically, FoV can be expressed as

$$FoV_{avg} = \frac{1}{NG_{vis}} \sum_{i=1}^{NG_{vis}} FoV_i \tag{23}$$

where NG_{vis} is the number of visited grids. Small value of FoV_{avg} means small frequency of revisiting a similar search site and vice versa. Notice that from equation (16), FoV = 1 means the robot never visited a similar grid more than once. Figure 9 shows the effect of l_{min} and μ on the FoV.

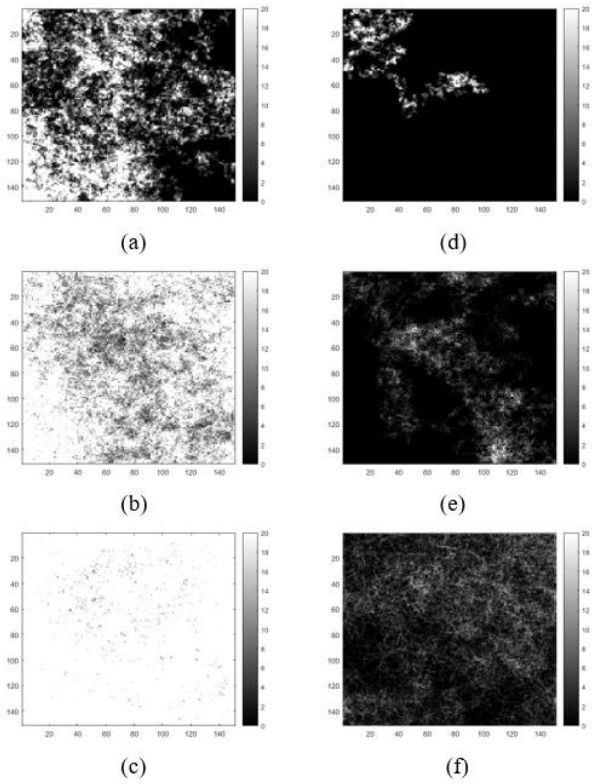


Figure 9 FoV for k = 10000 with (a) l_{min} = 1.5, μ = 1.1 (b) l_{min} = 15, μ = 1.1 (c) l_{min} = 50, μ = 1.1 and (d) l_{min} = 1.5, μ = 3.0 € l_{min} = 15, μ = 3.0 (f) l_{min} = 50, μ = 3.0

The average FoV for different values of l_{min}, μ and k_{max} is shown in Figure 10 for a n_A = 1. From the figure, the FoV increases as l_{min} increases or μ decreases. As evidenced from the figure, a large value of l_{min} results in a high possibility of revisiting the same area when μ is kept constant. This is because a large value of l_{min} means the robot may move farther from the current position due to the large size of minimum step length, l_{min}. As a result, for a fixed search space the potential of the robot to revisit the previously explored search space increases. A similar pattern is observed for different values of k_{max} when the same size of search space is considered.

For clarity, a heat map representing FoV for k_{max} = 100 is plotted as depicted in Figure 11. From the heat map, it can be clearly observed that a large value of l_{min} when μ is small gives a large value of FoV and vice versa. From the previous PAC analysis, μ = 1.1 is found to give optimal PAC when l_{min} is small. From the figure the average FoV is found to be lowest when l_{min} = 1.5 m when μ = 1.1. Thus, these values optimize PAC and at the same time minimize FoV which in this study are selected to be optimal parameters value of LF for source detection in a bounded search space.

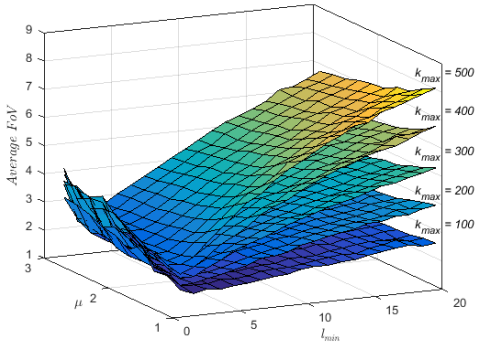


Figure 10 Average frequency of visiting for different values of l_{min}, μ and k for a search space 25×50 m² or n_A = 1

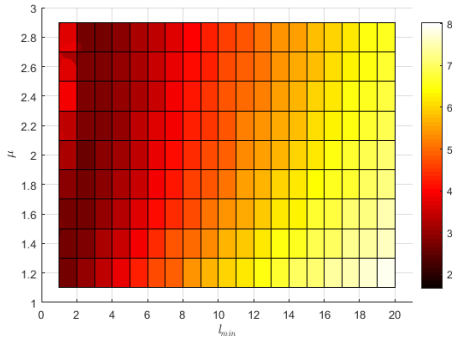


Figure 11 Heat map of average FoV for different values of μ and l_{min} for k_{max} = 100 iterations

In order to generalize this finding, a similar simulation is repeated for different sizes of search space. The average PAC for different values of l_{min} in different sizes of search space is shown in Figure 13. A similar pattern of performance can be observed for different sizes of search space but the corresponding value of PAC decreases as the size of search space increases. Thus, for any size of a bounded search space, movement with small value of μ performs better compared to the one with larger value of μ for small values of l_{min} and vice-versa. In addition, as the size of the search space increases, the optimal PAC value decreases for all values of μ where a small value of μ has better performance compared to a larger value of μ as shown in Figure 14.

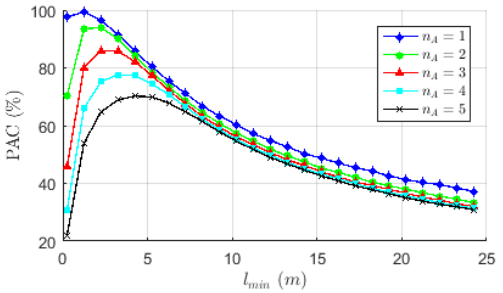


Figure 13 Average PAC for difference values of l_{min} and sizes of search space

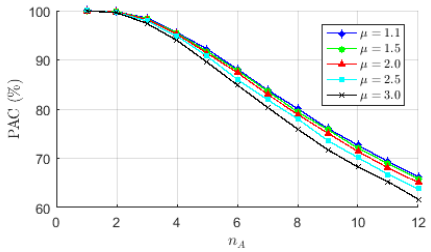


Figure 14 Area coverage for different sizes of search space for μ = 1.1 and l_{min} = 1.5 m and k_{max} = 1000

In the previous discussion, the optimal values of the LF parameters have been investigated for a single robot. The determined parameters value from the discussion are μ_{opt} = 1.1 and l_{min} = 1.5 m for a search space with size of 25×50 m^2. Based on these optimal parameters, the performance of the SDA algorithm can be further investigated for multiple robots. For the next simulation studies of the SDA, a similar search space of 25 × 50 m is considered. The relationship between the PAC value and number of robots and number of iterations is shown in Figure 15. From the figure, the PAC value increases as both the number of robots and number of iterations increases. For a small number of robots, a relatively high number of iterations is needed to achieve high PAC. On the other hand, using large a number of robots reduces the required number of iterations to achieve high PAC performance.

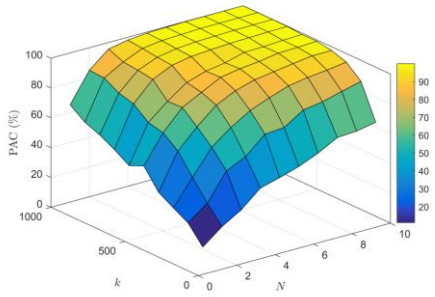


Figure 15 PAC for different number of robot and number of iteration for μ_{opt} = 1.1 and $l_{min,opt}$ = 1.5 m for 25 × 50 m2 unit search space for k_{max} = 1000

In this section, a complete DLF SDA performance at the swarm level is evaluated. Firstly, two factors that directly influence the detection performance of LF are the radius of detection, r_d (i.e., signal strength) and number of robots involved (i.e., swarm population size) as shown in Figure 20. From the figure, it can be observed that as the number of robots in the swarm increases, it significantly improves the source detection performance (i.e., the time taken to detect the source decreases). In addition, as the detection radius increases the detection time also decreases. However, as more and more robots are added to the swarm, the performance becomes saturated regardless of the detection radius.

Since it is known that LF performance is directly related to the detection radius of the source, to evaluate the impact of dispersion algorithm in DLF performance, the detection radius is fixed to r_d = 8 m and the size of search space is fixed to 25×50 m (i.e., n_A = 1). For the purpose of comparison, several benchmarking algorithms were selected. The comparison of the algorithm's performance for different population sizes of search space is shown in Figure 17. From the figure, DLF demonstrated a better performance especially for a small number of robots compared to benchmark algorithms for r_c = 20 m with r_{att} = 19 m, and r_{neu} = 16 m.

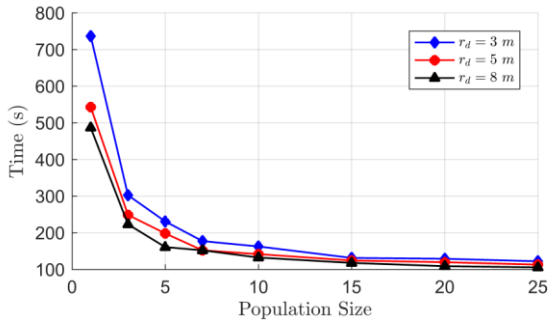


Figure 16 Time taken for LF to complete detection for different population size and different radius of detection for n_A = 1

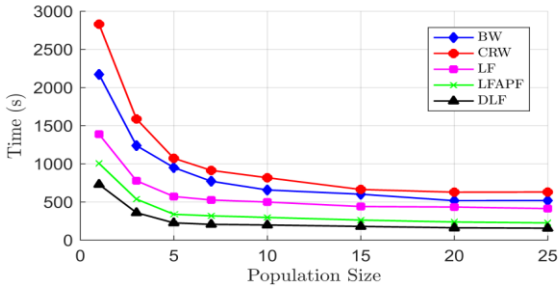


Figure 17 Comparison of time taken for different random detection algorithm for N = 5, n_A = 1, r_d = 8 m, r_c = 20 m with r_{att} = 19 m, and r_{neu} = 16 m

The Brownian walk (BW) is a pure random walk where its exploration is limited by its step's length. The correlated random walk (CRW) is similar to BW but its steps are correlated to the previous direction. The CRW has capability to drive direction of motion towards a certain direction if prior information about the source is available. However, in case the source is not detected, CRW will fail to perform efficiently. Lévy flight (LF) has capability of jumping to a new unexplored search space due to power law distribution of step lengths. As a result, its exploration capability is higher compared to BW and CRW. LF with artificial potential field (LFAPF) has capability of dispersing the robots in the search space. Consequently, LFAPF has improved detection time compared to LF. The proposed DLF algorithm has better performance compared to LFAPF by 124.5

seconds difference and other algorithms due to the fact that DLF has capability of repulsion to disperse the robots in the search space and remain neutral if robots are located at a sufficient distance from each other.

In addition, unlike LFAPF where robot dispersion is checked when robot updates its position (i.e., from t to $t + \Delta t$) which distorted the characteristic of the LF (i.e., exploration capability), DLF check for dispersion requirement every step update (i.e., from k to $k + 1$). Some other comparisons parameters are listed in Table 2 where DLF is clearly has better performance in term of highest area coverage at half of detection time (i.e., 78.6%), lowest detection time (i.e., 864.8 s with standards deviation of 113.1 s) and shortest average travel distance to reach detection (i.e., 166.1 m) compared to other algorithms. Thus, it is proven that the proposed DLF has better exploration and detection capability compared to other algorithms.

An example of a complete source detection execution is shown in Figure 18 for a source located at coordinate (20, 0). Starting from the same deployment site at (-23, -10), and since the source is not detected, robots perform a detection process until one of the robots detect the source. From the same region, robots were dispersed through a dispersion algorithm to optimize exploration using DLF where the communication connections among the robots are maintained as shown by magenta color lines. In this example, the time taken to complete the detection process is 138 s with 1380 step updates.

Table 2 Performance of DLF and benchmark algorithms

Algorithm	Detection Time (s)		Average Travel Distance (m)	Average PAC (%) at $t = t_{mean}/2$
	Mean	Standard Deviation		
DLF	864.8	113.1	166.1	78.3
Levy [26]	1431.1	201.3	193.8	56.6
LF+APF [27]	989.3	151.9	198.7	70.2
BW [28]	2215.7	325.5	288.6	31.7
CRW [28]	2814.1	356.2	315.8	35.1

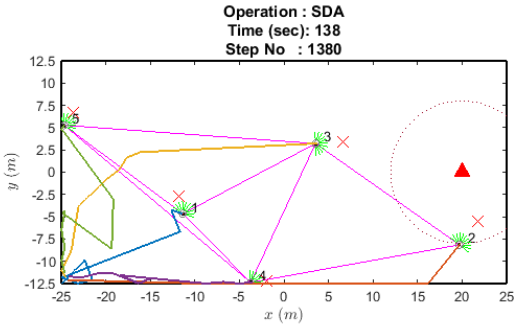


Figure 18 Robot trace during source detection with μ = 1.1, l_{min} = 1.5 m, r_c = 20 m, r_{att} = 19 m, r_{neu} ¬= 16 m, r_{rep} = 5 m and r_d = 6 m. Trace indicator: blue (R1), red (R2), yellow (R3), purple (R4), green (R5)

To evaluate performance of the proposed DLF in order to maintain communication connectivity during the source detection process, a number of neighborhood robots connected to a specific robot at a specific time is used. To show the reliability of the proposed DLF algorithm in ensuring communication connectivity, the average number of neighbors for each robot as time evolved is recorded. The idea is that if any robot fails to maintain communication connectivity with at least one of its neighbors, the neighborhood count will be zero. Figure 19 illustrates the connectivity among the robots during a 600 s of source detection period using the proposed DLF without attractive force implemented for the attractive radius for r_c = 20 m with r_{att} = 19 m, and r_{neu} - = 16 m. From the figure, one can clearly see that some robots have lost their connectivity with its neighbors for some period of time. Notice that since robots were deployed from the same initial position, they were initially connected to each other but as they explore the search space, some of them (i.e., R_1, R_3 and R_4 after approximately t = 430 s) start to lose communication with their neighbors in the absence of the ability to reconnect the network.

A similar simulation was repeated with attractive force components included to maintain a communication connectivity as shown in Figure 20. From the figure, it can be observed that the proposed attraction force successfully maintains communication network for the entire searching period where each robot maintains its connectivity with at least one of its neighbors. In this example, each robot is forced to stay within the communication radius in order to avoid communication losses by adjusting its position with respect to the nearest robot.

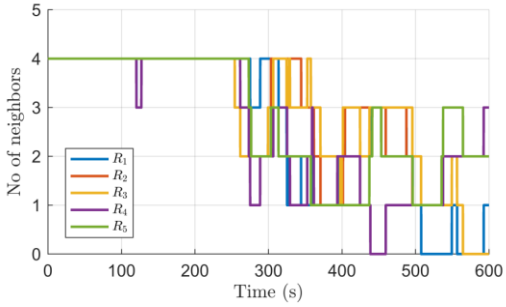


Figure 19 Communication connectivity among the robots without attraction force for r_c = 20 m with r_{att} = 19 m, and r_{neu} = 16

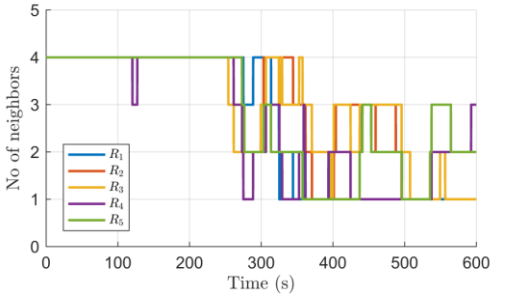


Figure 20 Communication connectivity among the robots with attraction force for r_c = 20 m with r_{att} = 19 m, and r_{neu} = 16 m

4.0 CONCLUSION

This paper presents a source detection algorithm based on an improved LF known as DLF. The LF is improved to ensure optimal area coverage and maintain communication among the robots in the swarm. To ensure the optimal area coverage, an artificial repulsive force is used to disperse the robots and avoid the robots search within the same area. To maintain communication and promote cooperative behavior among the agents, artificial attractive forces are proposed. Additionally, optimal LF parameters were identified through analysis of FoV and PAC. The results show that the proposed improvements performed better compared to existing or the benchmark algorithms. The DLF algorithm successfully improve detection time (113.1s) and area coverage (78.3%) compared to the existing algorithms: Brownian Walk (325.5s, 31.7%), Correlated Random Walk (356.2s, 35.1%), Levy Flight (201.3s, 56.6%), Levy Flight with Artificial Potential Fields (151.9s, 70.2%). In the near future, a complementary algorithm (i.e., tracing algorithm) utilizing the detected signal by DLF will be developed to complete the searching process. Additionally, a real-world experiment will be conducted using real hardware based on the framework proposed in this paper.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Acknowledgement

The authors would like to acknowledge the efforts, time, knowledge and facilities contributed by all parties from the related collaborative universities (i.e., UPSI. USM and UTEM) that make this research and publication possible.

References

[1] Senanayake, M., I. Senthooran, J. C. Barca, H. Chung, J. Kamruzzaman, and M. Murshed. 2016. Search and Tracking Algorithms for Swarms of Robots: A Survey. *Robotics and Autonomous Systems*. 75: 422-434. Doi: https://doi.org/10.1016/j.robot.2015.08.010.

[2] Wang, T.-M., Y. Tao, and H. Liu. 2018. Current Researches and Future Development Trend of Intelligent Robot: A Review. *International Journal of Automation and Computing*. 15(5): 525-546. Doi: https://doi.org/10.1007/s11633-018-1115-1.

[3] Sun, X., Y. Zhang, and J. Chen. 2019. High-level Smart Decision Making of a Robot Based on Ontology in a Search and Rescue Scenario. *Future Internet*. 11(11): 230. Doi: https://doi.org/10.3390/fi11110230.

[4] Ismail, Z. H., and M. G. M. Hamami. 2021. Systematic Literature Review of Swarm Robotics Strategies Applied to Target Search Problem with Environment Constraints. *Applied Sciences*. 11(5): 2383. Doi: https://doi.org/10.3390/app11052383.

[5] Majid, M. H. A., and M. R. Arshad. 2017. Cooperative Underwater Acoustic Source Searching based on Adaptive PSO Algorithm. *IEEE 7th International Conference on Underwater System Technology: Theory and Applications (USYS)*. Kuala Lumpur, Malaysia. 18-20 December 2017. 1-6. Doi: https://doi.org/10.1109/USYS.2017.8309449.

[6] Yang, J., R. Xiong, X. Xiang, and Y. Shi. 2020. Exploration Enhanced RPSO for Collaborative Multitarget Searching of Robotic Swarms. *Complexity*. 2020: 8863526-886338. Doi: https://doi.org/10.1155/2020/8863526.

[7] Yang, J., X. Wang, and P. Bauer. 2019. Extended PSO Based Collaborative Searching for Robotic Swarms with Practical Constraints. *IEEE Access*. 7: 76328-76341. Doi: https://doi.org/10.1109/ACCESS.2019.2921621.

[8] Dadgar, M., S. Jafari, and A. Hamzeh. 2016. A PSO-based Multi-robot Cooperation Method for Target Searching in Unknown Environments. *Neurocomputing*. 177: 62-74. Doi: https://doi.org/10.1016/j.neucom.2015.11.007.

[9] Lee, J. W., N. T. Tang, K. G. Lim, M. K. Tan, B. Yang, and K. T. K Teo. 2019. Enhancement of Ant Colony Optimization in Multi-Robot Source Seeking Coordination. *IEEE 7th Conference on Systems, Process and Control (ICSPC)*. Melaka, Malaysia. 13-14 December 2019. 200-205. Doi: https://doi.org/10.1109/ICSPC47137.2019.9068065.

[10] Zhang, X., and M. Ali. 2020. A Bean Optimization-based Cooperation Method for Target Searching by Swarm UAVs in Unknown Environments. *IEEE Access*. 8: 43850-43862. Doi: https://doi.org/10.1109/ACCESS.2020.2977499.

[11] Jiang, L., and P. Tian. 2021. A Bacterial Chemotaxis-Inspired Coordination Strategy for Coverage and Aggregation of Swarm Robots. *Applied Sciences*. 11(3): 1347-1366. Doi: https://doi.org/10.3390/app11031347.

[12] Peng, X., S. Zhang, and L. Xiaokang. 2016. Multi-target Trapping in Constrained Environments using Gene Regulatory Network-based Pattern Formation. *International Journal of Advanced Robotic Systems*. 13(5): 1729881416670152. Doi: https://doi.org/10.1177/1729881416670152.

[13] Huang, X. 2020. Improved 'Infotaxis' Algorithm-based Cooperative Multi-USV Pollution Source Search Approach in Lake Water Environment. *Symmetry*. 12(4): 549-567. Doi: https://doi.org/10.3390/sym12040549.

[14] Pablo, G.-A., C. Jaime del, and A. Barrientos. 2019. Behavior-based Control for an Aerial Robotic Swarm in Surveillance Missions. *Sensors*. 19(20): 4584. Doi: https://doi.org/10.3390/s19204584.

[15] Majid, M. H. A., and M. R. Arshad. 2017. A Combined Systematic and Metaheuristic Approach for Cooperative Underwater Acoustic Source Localization by a Group of Autonomous Surface Vehicles. *Indian Journal of Geo-Marine Sciences*. 46(12): 2434-2443. Doi: http://nopr.niscpr.res.in/handle/123456789/43189.

[16] Pang, B., Y. Song, C. Zhang, and R. Yang. 2021. Effect of Random Walk Methods on Searching Efficiency in Swarm Robots for Area Exploration. *Applied Intelligence*. 51(7): 5189-5199. Doi: https://doi.org/10.1007/s10489-020-02060-0.

[17] Dimidov, C., G. Oriolo, and V. Trianni. 2016. Random Walks in Swarm Robotics: An Experiment with Kilobots. *Swarm Intelligence. ANTS 2016*. In Lecture Notes in Computer Science. Springer, Cham. Doi: https://doi.org/10.1007/978-3-319-44427-7_16.

[18] Ramachandran, R., Z. Kakish, and S. Berman. 2020. Information Correlated Lévy Walk Exploration and Distributed Mapping Using a Swarm of Robots. *IEEE Transactions on Robotics*. 36(5): 1422-1441. Doi: https://doi.org/10.1109/TRO.2020.2991612.

[19] Zaburdaev, V., S. Denisov, and J. Klafter. 2015. L'evy Walks. *Reviews of Modern Physics*. 87(2): 483-530. Doi: https://doi.org/10.1103/RevModPhys.87.483.

[20] Katada, Y., *et al.* 2016. Swarm Robotic Network using Lévy Flight in Target Detection Problem. *Artif Life Robotics*. 21: 295-301.
Doi: https://doi.org/10.1007/s10015-016-0298-1.
[21] Khaluf, Y., S. Van Havermaet, and P. Simoens. 2018. Collective Lévy Walk for Efficient Exploration in Unknown Environments. *18th International Conference on AIMSA. Varna, Bulgaria*. 12–14 September 2018. 260-264.
Doi: https://doi.org/10.1007/978-3-319-99344-7_24.
[22] Majid, M. H. A., and M. R. Arshad. 2019. Search Space Exploration using Lévy Flight with Turning Angle Constraint and Boundary Reflection. *22nd International Conference on Climbing and Walking Robots and Support Technologies for Mobile Machines*. Kuala Lumpur, Malaysia. 26-28 August 2018. 133-140.
Doi: https://doi.org/10.13180/clawar.2019.26-28.08.18
[23] Palyulin, V. V., A. V. Chechkin, and R. Metzler. 2014. Lévy Flights do not Always Optimize Random Blind Search for Sparse Targets. *Proceedings of the National Academy of Sciences.* 111(8): 2931-2936.
Doi: https://doi.org/10.1073/pnas.1320424111.
[24] Majid, M. H. A., and M. R. Arshad. 2016. Design of an Autonomous Surface Vehicle (ASV) for Swarming Application. *IEEE/OES Autonomous Underwater Vehicles (AUV)*. Tokyo, Japan. 6-9 November 2016. 230-235.
Doi: https://doi.org/10.1109/AUV.2016.7778676.
[25] E. W. McGookin. 2001. AUV Sliding Mode Autopilot Optimisation Using Genetic Algorithms. IFAC Proceedings. 34: 317-322.
Doi: https://doi.org/10.1016/S1474-6670(17)35102-9.
[26] Keeter, M., D. Moore, R. Muller, E. Nieters, J. Flenner, S. E. Martonosi, A. L. Bertozzi, A. G. Percus, and R. Levy. 2012. Cooperative Search with Autonomous Vehicles in a 3D Aquatic Testbed. *American Control Conference (ACC). Montreal, QC, Canada.* 27-29 June 2012. 3154-3160.
Doi: https://doi.org/10.1109/ACC.2012.6314965.
[27] Sutantyo, D. K., S. Kernbach, P. Levi, V. A. Nepomnyashchikh. 2010. Multi-robot Searching Algorithm using L´evy Flight and Artificial Potential Field. *IEEE Safety Security and Rescue Robotics*. Bremen, Germany. 26-30 July 2010. 1-6.
Doi: https://doi.org/10.48550/arXiv.1108.5624.
[28] Cao, M.-L., M. Hao, B. Luo, and M. Zeng. 2015. Experimental Comparison of Random Search Strategies for Multi-robot based Odour Finding without Wind Information. *Austrian Contributions to Veterinary Epidemiology*. 8: 43-50.
Doi: https://doi.org/10.5281/zenodo.33822.