International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Invasive Weed and Grasshopper Optimization based on AI Approach for Enhanced Routing in FANETs.

Ch. Naveen Kumar Reddy[1], Dr. M. Anusha[2]

Submitted: 30/12/2023 **Revised**: 06/02/2024 **Accepted**: 14/02/2024

Abstract*:* Methods of clustering show promise as instruments for ensuring the scalability and maintainability of massive FANETs. However, it is challenging to uphold the FANETs because of Unmanned Aerial Vehicles (UAVs). FANETS routing is more difficult than MANETs or VANETs because of these topological constraints. When static and dynamic routings aren't enough to fix a complex routing problem, clustering methodologies based on AI can be employed to find a solution. This paper proposes a method for solving such routing issues by incorporating the benefits of the Invasive Weed Optimization Algorithm (IWOA) into the Grasshopper Optimization Algorithm (GOA). This method is referred to as Hybrid Invasive Weed Improved Grasshopper Optimization Algorithm-based efficient Routing (HIWIGOA-R). In particular, the random walk tactic is used to avoid the potential for a single solution to dominate. The traditional GOA's exploitation coefficient was adjusted through the use of grouping to achieve a more equitable rate. The effectiveness of the suggested approach is measured in a variety of ways. These include packet delivery ratio, end-to-end delay, energy consumption and network lifetime. The experimental consequences presented here show that the suggested algorithm outperforms the current top methods in the field.

Keywords: *Flying ad hoc networks, Clustering, Routing, Invasive weed optimization algorithm, Grasshopper optimization algorithm*

1. Introduction

In the area of wireless communication and network technology, FANETs (Flying Ad Hoc Networks), often referred to as MANET (Mobile Ad-hoc Networks) in an aerial setting, is an exciting and quickly developing topic [1]. These networks, which are made up of drones or UAVs, have drawn a lot of attention lately because of their adaptability and use in a variety of contexts, such as precision agriculture, disaster relief, military operations, and even entertainment. Sophisticated routing protocols are the foundation of any FANET and are necessary to facilitate communication between these flying nodes [2]. FANET, is a subset of (MANETs) designed specifically with Unmanned Aerial Vehicles (UAVs) in mind. FANETs are very dynamic and self-configuring, and they don't have a set infrastructure as conventional networks have. These networks are made up of a fleet of unmanned can talk to one additional and carry out a function [3]. These responsibilities might include information relaying, data collecting, monitoring, and surveillance in settings where traditional networks are impracticable or not possible. Because of its prospective uses, FANETs have piqued the interest of academicians, engineers, and inventors [4]. They provide answers to problems that arise in the real world, such managing urban traffic, environmental monitoring, precision agriculture, and disaster response. Because FANETs don't need pre-existing infrastructure and can be deployed on-demand, they are especially appealing in situations where flexibility and quick deployment are crucial.

Establishing wireless connection amongst drones—which are often mobile and dispersed over large areas—is the basic idea behind FANETs. Because of this, effective routing procedures are required to guarantee dependable and prompt data transmission [5]. The drones use routing protocols as their navigational compass to find the optimal routes for exchanging messages and data. In order to fully use FANETs in practical applications, these routing protocols must be functional [6]. The mobility of the nodes, or drones, is one of the main issues with FANET routing. UAVs can move in three dimensions and continually change positions. Due to this mobility, network topologies become more dynamic, with nodes constantly entering and leaving communication areas. Because they rely on comparatively stable network topologies, traditional routing techniques often can't keep up with the high dynamism of FANETs [7]. It is a challenging task to guarantee dependable data transmission while making adjustments for frequent topological changes. Due to limitations on their communication gear and the intrinsic characteristics of the wireless channel, drones in FANETs usually have short communication ranges [8].

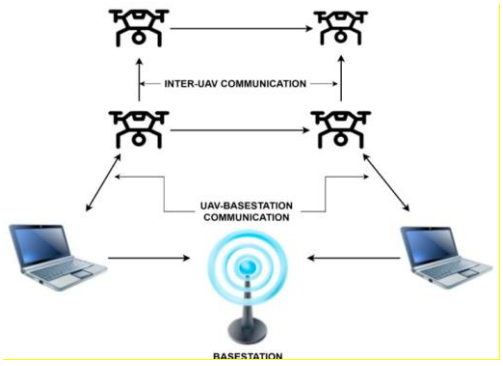

Fig 1. FANETS Network

1 Research Scholar, Dept. of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, AP, India
ORCID ID : 0000-0002-9801-6400
naveenkumarreddy.ch@gmail.com
2 Assoc.Professor, Dept. of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, AP, India,3 Computer Eng.
anushaaa9@kluniversity.in

It is difficult for nodes to establish direct communication linkages with far-off neighbors because of this limitation. This leads to the widespread use of multi-hop communication, in which data is sent through intermediary nodes [9].

To guarantee that data reaches its destination, routing methods need to identify these intermediary nodes effectively. Data security and privacy are issues that are brought up by the usage of drones for a variety of purposes, such as data collecting and monitoring [10]. These concerns must be taken into account by FANET routing protocols in order to provide safe data broadcast and network resilience against any intrusions or unauthorized access [11]. Wi-Fi networks, cellular networks, and other FANETs are among the wireless communication technologies that share the electromagnetic spectrum with FANETs. Maintaining the quality of service and preventing communication breakdowns require effectively controlling interference and making use of the available spectrum [12]. Novel routing protocols that are customized to the particular needs of FANETs are needed to address these issues. Scientists and engineers have been hard at work creating and improving these protocols in order to get over the challenges presented by the aerial ad hoc networking environment [13].

Remaining units of the paper are organized as: In Section 2, the essential literatures are reviewed, and in Section 3 and 4, the proposed model is briefly discussed. The results and an overview of the validation process are included in Section 5. A conclusion is provided in Section 6 to finish.

2. Related Works

The problems of delay and packet loss that plague existing routing protocols in highly dynamic topology settings were explored in depth in a recent study by Xue, Q., et al. [14]. For mobile ad hoc networks, it suggested a Q-learning-Enabled, Routing Algorithm(QEHLR). This research used a Q-learning approach to learn network and successfully pick routes to prevent connection loss, as opposed to the traditional routing algorithms that were ineffective in highly dynamic FANETs. In addition, the routing protocol incorporated the residual time of the connection or path's lifespan into the routing table's construction. QEHLR eliminated links it foresees gets failed based on the current state of the network, cutting down on packet loss due to incorrect route selection. In order to solve the problem of routing methods being unable to handle a wide variety of mobility scenarios in FANETs with a lively topology, the authors provided a control factor protocol, which, as simulations indicated, greatly increased the packet transmission rate. The experimental data showed that by using the new routing technique, network performance could be greatly enhanced.

QoS may be enhanced by the management of UAVs' energy consumption, congestion, and geolocation data, as suggested by the research of Anwekar, D., and Phulre, S. [15]. Although many aspects affect quality of service enhancement, this study focused on energy, congestion, physical location, UAV size, and UAVs' significance in agriculture. Minimal-cost routes were picked adaptively depending on current network conditions, and routing protocols and procedures were tweaked or updated as necessary to maintain a high ratio of successfully delivered packets and traditional routing processes. Research into routing protocols was also covered to provide light on current efforts in the field and point the way towards future improvements.

Details of machine learning-based intrusion detection systems were the subject of Rahman, K., et al.'s [16] research. The UNSW-NB 15 dataset was also used to model a cognitive lightweight-LR strategy. Using machine learning, the paper developed an IoT-based UAV network that can identify potential security breaches. In the network, the work used regression. The proposed method for estimating statistical likelihood is logistic regression. The binomial distribution was the standard for scientific investigation. In logistic regression, there was a linear association method. Logistic regression was an inexpensive and very lightweight method. Logistic regression was shown to perform better than alternative methods in the simulations. High precision was also optimally matched.

Fuzzy FANETs, as proposed in the study by Hosseinzadeh, M., et al. [17]. FTSR used two different types of trust evaluation mechanisms: local trust. The local trust approach was a decentralised method for identifying trustworthy nearby nodes and excluding malicious ones from the network. As such, only trusted nodes were permitted to take part in the path finding process. In FANETs, this reduced the possibility of forged connections being established. When malicious nodes were not discovered during the local trust process, the path trust strategy was tasked with finding them. This tactic provides an overview of the reliability of the preferred route. This approach was designed using a fuzzy scheme operating at the source node to choose the most secure path among the source and the destination. Final findings including malicious detection latency were achieved when FTSR was implemented in Network Simulator 2 (NS2). These findings suggested that FTSR outperformed TOPCM, MNRiRIP, and MNDA. However, FTSR was slower than TOPCM when it came to locating viable pathways.

Using UAVs fused with Wireless Body Sensors (WBSN), the study by Kumar, S., et al. [18] suggested an emergency info dissemination protocol called SF-GoeR in FANET to instantly transfer the patient's current health info to the neighbouring hospital system. By taking essential factors of UAVs—closeness ratio and residual energy ratio—the projected SF-GeoR aimed to improve the flow of emergency information among medical personnel and patients while decreasing the frequency with which the links between them get disconnected. Key measures network longevity, and latency were used to compare the SF-GoeR's performance to that of the current GPSR-WG and GPSR methods.

A routing technique for FANETs was proposed in the study by Kumar, S. et al. [19]. By making educated guesses about aimed to increase the connection duration among the UAVs in scenarios where a source chosen a forwarding UAV from a specified set of neighbours. The proposed LoCaL reduced route breaks between the source and the destination, making the connection more reliable. In addition, the utility function was used to mathematically formulate the suggested technique, which selected all relay UAVs inside the cone-shaped request zone to increase route stability while decreasing routing overhead during route discovery. Key metrics network lifespan, and latency have been provided to sum up the LoCaL's performance in comparison to the current methods.

An energy-aware routing system was presented for FANETs by Lansky, J., et al. [20]. The idea for this protocol came from OLSR, or optimised link state routing. The suggested routing strategy makes use of a novel technique that makes use of two parameters:

the ratio of sent/received of hello packets and the evaluate among two mobile nodes in the air. The firefly algorithm was also used to decide which Multipoint Relays (MPRs) to use in the suggested technique. MPR was assigned to the node with the highest sum of residual energy, highest quality of connections, largest number of neighbours, and largest willingness. Finally, their plan proposed a method for generating routes between nodes that optimised energy consumption and network quality. By testing the suggested routing method on the NS3. When compared to greedy Optimised Link State Routing (OLSR), its simulation results were deemed superior. When compared to G-OLSR and OLSR, the technique had better latency, packet delivery rate, throughput, and energy consumption outcomes. However, in comparison to G-OLSR, the routing overhead in the suggested technique was significantly higher.

2.1. Research Gap

Numerous significant research advances in the area of routing protocols for extremely dynamic FANETs were presented in the literature currently in publication. One area of neglected research need is that, although the QEHLR routing algorithm which uses Q-learning to enhance route selection and reduce packet loss no thorough assessment or comparison with other current routing protocols has been done to confirm the algorithm's efficacy in a range of mobility scenarios. Additionally, there was research on vacuum concerning the incorporation of machine learning-based intrusion detection systems into FANETs and their possible effects on network security. Furthermore, as discussed in research, there is an uncharted territory with regard to energy-efficient routing strategies for FANETs that take into account the unique constraints and requirements of UAVs. The research presented in the concluding paragraph presents a novel HIWIGOA-R approach, but it lacks a comparative analysis with existing clustering and routing techniques in FANETs, despite the fact that clustering methodologies based on AI have shown promise. By filling in these gaps, can improve knowledge and create more effective routing solutions for highly dynamic FANETs.

3. Preliminaries

The HELLO message's structure and the study's optimisation objective will be presented below.

3.1 System Model

Figure 2 depicts the system paradigm, in which all UAVs are controlled remotely via the cloud. The cloud creates clusters and updates the routing table used by the CHs to communicate. The cloud sends out command signals to all UAVs via CHs

All UAVs in this system have location-aware subsystems and wireless communication interfaces built in, allowing them to track their own velocity, heading, and coordinates, as well as follow predetermined routes. The FANET will be centralized on the cloud. A CH and CMs make up each cluster. Once the clusters have been formed, the CHs within each cluster partition the grid according to their respective comm ranges. The HELLO message allows the CH to learn the location and velocity of CMs through their interaction. To conserve power and extend the life of the system, the CH modifies the strength of its signal transmission based on the location of the CMs in the communication grid. The CH also makes predictions about CM behavior, makes judgements about where CMs will be in the future, and makes timely adjustments to the signal's transmission strength. To keep things running smoothly and prevent any disruptions in communication or the network's longevity due to a single node's excessive energy consumption, they reorganize clusters and elect Cluster Heads (CHs) using a weighted summation-based procedure.

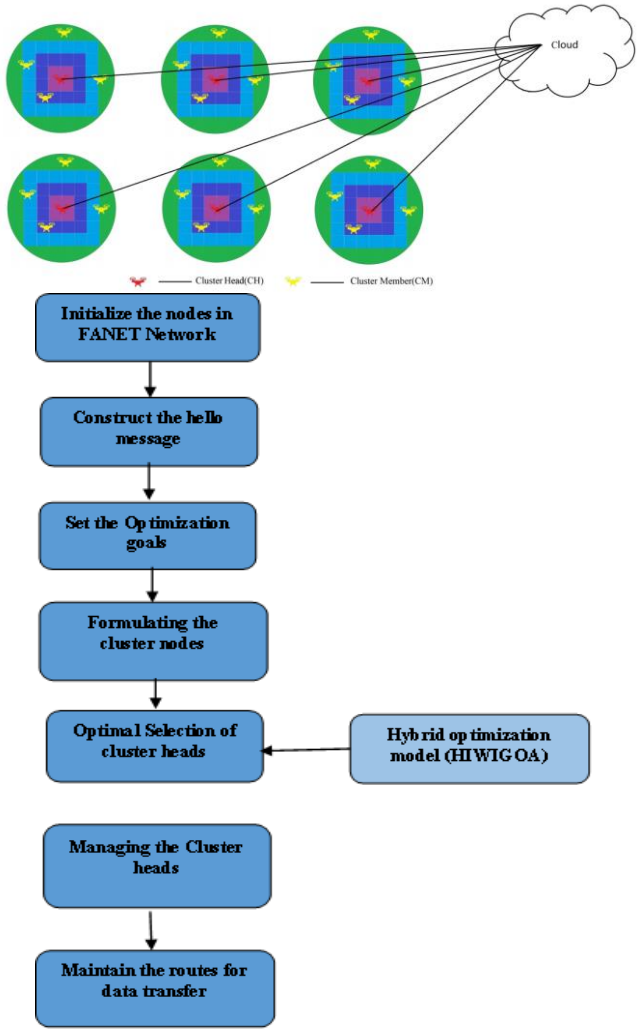

Fig 2. The System Model.

3.1 HELLO Message

Through HELLO message exchange, UAVs learn about neighboring nodes for clustering purposes. Figure 3 depicts the HELLO message's structure.

The subsequent is what is included in a HELLO message:

- ID: the identification of the UAV node.
- Cluster ID: the identification of the cluster.
- Role: the value can be 0 or 1, where 0 represents CH and 1 represents CM.
- Weight: the weight of a CH candidate in the CH selection phase.
- Speed: the speed of the UAV node.
- Direction: the flying direction angle of the UAV node.
- Position: the site of the UAV node

ID	Cluster ID	Role	Weight	Speed	Direction	Position

Fig 3. HELLO message construction.

3.3. Optimization Goal

3.3.1. The reliability of the FANET

The network's dependability may be gauged by looking at the PDR. Equation (1) characterizes the PDR as

$$PDR = \frac{P_R}{P_S} \tag{1}$$

where P_R characterizes the sum of data packets successfully conventional by the terminus nodes, and P_S characterizes the total sum of data packets produced by the s nodes.

The more the PDR, the clearer the gearbox. This study presents a technique for electing a cluster head and a method for keeping the cluster running smoothly that together maximise PDR. The cluster head election technique will choose the most reliable CHs to forward packets, increasing the network's chance of success. UAV connectivity and transmission success rate are both enhanced by efficient cluster maintenance, which decreases the likelihood of link disconnection.

3.3.2. The Lifetime of the FANET

When a node's outstanding energy drops below 20% of its starting value, it is considered to be dead and is removed from the FANET [21]. The efficiency of the FANET will suffer if the number of inoperable nodes is high. Therefore, the network will be deemed inoperable when half of its nodes have died. Thus, the FANET lifespan may be calculated using equation (2).

$$T = t_e - t_s \tag{2}$$

where t_s is the period when the t_e characterizes the time invalid.

CHs do more work and use more energy than CMs do. Consequently, the longevity of the FANET may be significantly extended by decreasing the energy required by CH. By relying on the method that allows CHs to dynamically modify their transmitting power, the CH may save energy by tailoring its transmission to the distance to each CM. This reduces the pace at which nodes die and extends the useful life of the FANET.

3.4. Network Model

Assume that N UAVs are randomly dispersed across a 22500 m area, that each UAV has a unique ID, that node locations are (x,y,z), and that the base station's E_i, and that the sensing radius R and communication radius r of each UAV is 250 m and 300 m, respectively. For improved efficiency and reduced power consumption, must now cluster N UAVs into K UAVs., and $U=\{UAV_1, UAV_2,...,UAV_i,...,UAV_N\}$ is the set of all UAV nodes, $\{C=CH_1,CH_2,...,CH_j,...,CH_k\}$ is the set of all CH nodes.

3.5. Energy Model

Energy needed for flight and hovering, communication with other UAVs, and sensor operation all contribute to the total FANET, which is represented by equation (3).

$$E_T = E_H + E_F + E_C + E_S \tag{3}$$

UAV flight and hover determined using the formula in Equation (4), which is based on [15]. Here is how the hovering power of a UAV is determined:

$$PH = \sqrt{\frac{(m_u g)^3}{2\pi r_w^2 n_w \rho_a}} \tag{4}$$

where m_u is the UAV mass, g is the gravity, r_w and n_w symbolize the radius and sum of wings, correspondingly, and r_a is the air density.

The UAV flight power is intended in equation (5) as shadows:

$$P_F = (P_{max} - P_H)\frac{v_u(t)}{v_{max}} \tag{5}$$

where $v_{\max}$ is the extreme UAV flight speed, $v_u(t)$ is the UAV $v_{\max}$

The UAV ingesting is mentioned in equation (6) as shadows:

$$\begin{cases} E_H = P_H T_H \\ E_F = \int_0^{T_F} P_F dt \end{cases} \tag{6}$$

where T_H and T_F characterize the hovering and flying time, correspondingly. Referring to [16], the channel found perfect is chosen according to the distance among nodes and the energy required for UAV data transmission. To determine how much power a node needs to send l bits of data over a given distance d, the formula is expressed as:

$$E_{TX}(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & d < d_t \\ lE_{elec} + l\varepsilon_{amp}d^4 & d \geq d_t \end{cases} \tag{7}$$

where E_{elec} denotes the transmittal l bits of data, ε_{fs} and represents the power amplifier's energy consumption characteristics in the free- distance threshold, which may be found by solving equation (8).:

$$d_t = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{amp}}} \tag{8}$$

The energy spent to receive l bit data is intended in equation (9) as shadows:

$$E_{RX} = l \times E_{elec} \tag{9}$$

4. Scheme Description

4.1. Optimal Quantity of Cluster Heads

Number of clusters correlates with total network energy usage. Higher CH node load, earlier energy depletion, and more frequent are the results of having too few clusters. However, the number of routing hops grows when too many clusters are divided. Increases in transmission latency are the result; establishing the best cluster size helps cut down on wasted resources and power. In this research, a coverage and bandwidth balance were used to calculate the optimum number of clusters.

4.1.1. Coverage Analysis

In this study, an improved version of the coverage analysis technique described in [7] was provided. UAV_iU and CH_jC are said to be within a CH's perceptual sphere if and only if the Euclidean distance between UAV node i and CH node j perception radius. Typically, a UAV node will associate with the CH UAV that is geographically closest to it. The link state between nodes UAV_i and UAV_j is characterized by L_{ij} in the binary perception model. If D_{ij}, the closest distance from UAV node i to CH, is less than or equal to the CH sensing radius, then UAV node i and CH are in sensing range of each other., set L_{ij}=1; otherwise, L_{ij}=0.

$$D_{ij} = min\{Dis(i,1), Dis(i,2), .., Dis(i,K)\} \tag{10}$$

$$Dis(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{11}$$

$$L_{ij} = \begin{cases} Dis(i,j) \le R_j \\ Dis(i,j) > R_j \end{cases} \quad (12)$$

In order to achieve the desired result of having all nodes participate in the cluster structure, it is required that each node be under the protection of at least one CH. In contrast, only one cluster can include a given UAV node, meaning that the following equations (13) and (14) must be satisfied by the total number of CHs.:

$$\sum_{\forall j \in C} L_{ij} = 1, \ \forall i \in U \quad (13)$$

$$\sum_{\forall i \in U} L_{ij} = N - K \quad (14)$$

4.1.2. Bandwidth Analysis

When the number of clusters is too low, the sum of nodes in the cluster grows, increasing the likelihood of lowering throughput; when the number of clusters is too high, increasing throughputs should be equalized for optimal network utilization. Where M_j is the total sum of CMs in the j^{th} bandwidths, respectively. The bandwidth limits in equations (15) and (16) should be used to regulate the ideal number of CHs:

$$\frac{1}{K}\sum_{j=1}^{K} \frac{B_1}{\sqrt{M_j}} \le \frac{B_2}{\sqrt{K}} \quad (15)$$

$$\sum_{\forall j \in C} M_j = N - K \quad (16)$$

By calculating the goal function in equation (17), the ideal number of CHs were determined to solve.:

$$\begin{aligned} & min \sum_{j=1}^{K} j \\ s.t. \ & \sum_{\forall j \in C} L_{ij} = 1, \forall i \in U \\ & \sum_{\forall j \in u} L_{ij} = N - K \\ & \frac{1}{K}\sum_{j=1}^{K} \frac{B_1}{\sqrt{M_j}} \le \frac{B_2}{\sqrt{K}} \end{aligned} \quad (17)$$

Constraints: 1. Each UAV in the cluster is linked to a single CH, and all CHs are linked to each other. Third, bandwidth is evenly distributed both inside and among clusters.

4.2. Cluster Formation

Once the sum of clusters has been determined, the Binary Whale Optimisation (BWOA) algorithm is used to make a real-time decision on who will serve as the cluster's leader based on criteria balancing. After that, the distance between nodes were used to group the UAVs into the most nearby clusters.

4.2.1. CH Selection Based on IGOA

4.2.1.1. Standard Grasshopper Optimization Algorithm (GOA)

Equation (18) is used in the normal GOA to regulate how to inform the site ($SA_{G(i)}$) of the grasshoppers (appropriate VMs.

$$SA_{G(i)} = S_{I(i)} + G_{F(i)} + W_{A(i)} \quad (18)$$

where $S_{I(i)}$ and $G_{F(i)}$ characterize the social change. It also represents the wind that determines how much exploration or exploitation the search agent needs to do. The social interaction factor is now calculated using equations (19) and (20)

$$S_{I(i)} = \sum_{j=1.j\neq i}^{S_N} s(d_{ij}) \frac{(x_j - x_i)}{d_{ij}} \quad (19)$$

$$s(r) = ae^{-\frac{r}{b}} - e^{-r} \quad (20)$$

where "a" and "b" stand for the modifying elements that provide a foundation for achieving adaptability in social settings. In addition, using equation (21), can get the value of d_{ij}, which stands for the Euclidean distance with regard to the i_{th} and j_{th} search agents.

$$d_{ij} = |x_j - x_i| \quad (21)$$

In addition, using equations (22) and (23), the social interaction was determined and Gravitational force parameters in relation to the search agent.

$$G_{F(i)} = -g_{const} \widehat{v(e_g)} \quad (22)$$

$$W_{A(i)} = -u_{const} \widehat{v(e_W)} \quad (23)$$

where, g_{const} and u_{const} characterize the constants, in the search agents. Also, $\widehat{v(e_g)}$ and $\widehat{v(e_W)}$ force, correspondingly, emphasizing the implication of the search agent drive updating procedure.

Also, the search agent updating procedure obtainable in equation (24) by including the parameters of $S_{I(i)}, G_{F(i)}$ and $W_{A(i)}$ established through equations (19), (22), and (23).

$$SA_{G(i)} = \sum_{j=1,j\neq i}^{S_N} s(d_{ij}) \frac{(x_j - x_i)}{d_{ij}} - g_{const} \widehat{v(e_g)} + u_{const} \widehat{v\ (e_w)} \quad (24)$$

If counting N grasshoppers, then. The following is a reformulated mathematical model of equation (25) that may be used to use the GOA to solve the optimisation challenges:

$$SA_{G(i)} = \sum_{j=1,j\neq i}^{S_N} s(d_{ij}) \frac{(x_j - x_i)}{d_{ij}} - g_{const} \widehat{v(e_g)} + u_{const} \widehat{v(e_W)} \quad (25)$$

whereby S_N emphasises the total number of grasshoppers (search agents). Tasks are assigned to the most suitable Virtual Machines (VMs) according to the equation (26), which is a modification of the mathematical model used in this optimisation.

$$SA_{G(i)} = c\left(\sum_{j=1,j\neq i}^{S_N} c \frac{U_T^d - L_T^d}{2} s\left(x_j^d - x_i^d\right) \frac{(x_j - x_i)}{d_{ij}}\right) + \widehat{P_{Best}} \quad (26)$$

where U_T^d and L_T^d depict the dimension "d" exploration, with $\widehat{P_{Best}}$ being the agent during the procedure of optimization. In count, the value is strongminded based on equation (27).

$$c = c_{Max} - Iter_{Curr}\left(\frac{c_{Max} - c_{Min}}{Iter_{Max}}\right) \quad (27)$$

where c_{Max} and c_{Min} characterize the least and extreme standards of the adjusting constant, respectively, and $Iter_{Curr}$ and $Iter_{Max}$ signify the extreme sum of repetitions and the current application iteration.

4.2.2 Inclusion of IWOA for Refining GOA

While each search agent is in motion in the GOA procedure, its function value is disregarded. A larger step is required by the search agent with a higher value of impartial function to find the optimal keys that can be located in the search space. This means that GOA's exploitation potential is very lacking and in need of development. The issue of being stuck at a "local point of optimality" is also present in GOA. IWOA is included into the GOA algorithm to achieve improved VM resource allocation and systematic load balancing, thereby solving these two problems. In order to achieve global optimality in the load balancing and task scheduling processes, the HIWIGOA-LB strategy is developed. The exploitation potential of this IWOA/GOA hybrid was

increased by the use of a random walk method. By managing the search agents' incremental progress towards the optimal solution, IWOA helps GOA converge more quickly. Furthermore, the goal function values, and iterative numbers are employed to achieve the kinds and steps needed by the HIWIGOA for position updating. In particular, the local search capability is enhanced by first hybridising the random way and IWOA algorithms. Second, the trade-off among exploration is mitigated by employing the grouping technique. As shown in Fig. 1, the HIWIGOA algorithm is a mix of IWOA and GOA.

In this case, the IWOA algorithm's search agent (plant) can provide better optimum solutions if it has more robust objective functions. Equation (28) displays the maximum number of solutions that may be created by the IWOA process.

$$Ps = \frac{Ob_{Fn(i)} - Obc_{Fn(warsti)}}{Ob_{Fn(Best)} - Obc_{Fn(warsti)}}\left(Ps_{(Max)} - Ps_{(Min)}\right) + Ps_{(Min)} \quad (28)$$

where; $Ps_{(Max)}$ and $Ps_{(Min)}$ characterize that can be perhaps produced by the IWOA procedure. In specific, $Ob_{Fn(Best)}$ and $Obc_{Fn(warsti)}$ highlight the function standards strongminded by the procedure. The solutions based on equation (29).

$$\sigma_{Iter} = \left(\frac{(Iter_{Max--Iter_{Curr}})^n}{(Iter_{Max})^n}\right)(\sigma_{Init} - \sigma_{Final}) + \sigma_{FInal} \quad (29)$$

where, σ_{Init} and σ_{Final} characterizes the preliminary and final values related with the generated candidate solution.

(A) Strategy of Random Walk

In this projected HIWIGOA procedure, the plan of enhancing the during the procedure of available VMs. In this policy, novel solutions ($SA_{New\_G(i)}$) are strongminded primary ($SA_{FBEST\_G(i)}$) and secondary finest solutions ($SA_{SBest\_G(i)}$) as stated in equation (30).

$$SA_{New_{G(i)}} = Rand * SA_{FBEST_{G(i)}} + Rand * SA_{SBest\_G(i)} \quad (30)$$

(B) Strategy of Grouping

Optimization algorithms can benefit from the employment of a variety of inertial weights to maximize their performance in specific settings. This clustering method is applied to the classic GOA's coefficient c to adjust coefficient c is seen to decline more linearly with increasing iterations. It's goal is to increase the likelihood of progress. The value of the adjusted coefficient is then calculated using equation (31).

$$c_{mod} = c_{Min}\left(\frac{c_{Max}}{c_{Min}}\right)^{\frac{1}{1+A}} const\left(\frac{Iter_{Curr}}{Iter_{Max}}\right) \quad (31)$$

The coefficient, c_{mod}, is utilized for search procedure optimization. Using the values of their goal functions as determined by equation (32), the entire population of search agents is divided into three agents.

$$c_{mod} = c_{Max} - Iter_{Curr}\left(\frac{c_{Max} - c_{Min}}{Iter_{Max}}\right) \quad (32)$$

At this stage, the best search agents are taking baby steps forward in their pursuit of an improved goal function. While the GOA algorithm utilizes large values of function and targets to update its location, the observer search agent employs moderate values. In particular, the classic GOA algorithm's inclusion of the problem of local optimality is mitigated with the use of scout search agents. The poorest scout search agents' starting positions are randomly selected for the first three quarters of this strategy's iterations. The scout search agent is crucial to the enhancement of exploration capabilities throughout the remaining 25% of iterations. Equation (33) is used to adjust the coefficient (c_{mod}) that is linked to the scout search agent.

$$c_{mod} = c_{Max} - (c_{Max} - c_{Min})\left(\frac{Iter_{Curr}}{Iter_{Max}}\right)^4 \quad (33)$$

Last but not least, this improved (c_{mod}) version is comprised into equation (24) to brand it optimal for transmission incoming tasks to appropriate VMs, as shown in equation (34).

$$SA_{G(i)} = c_{mod}\left(\sum_{j=1,j\neq i}^{S_N}(c_{mod})\frac{U_T^d - L_T^d}{2}s\left(x_j^d - x_i^d\right)\frac{(x_j - x_i)}{d_{ij}}\right) + \widehat{P_{Best}} \quad (34)$$

4.2.2. Cluster Management Stage

As the UAV nodes use energy and move, it's possible that the remaining nodes in the cluster won't be enough to serve as cluster chiefs once the first split is complete.

The following scenarios need for regular cluster upkeep:

i. Establish a minimum acceptable node energy and do routine checks on node energy levels for CH. If it drops below this threshold or CH quits the cluster, cluster maintenance must be performed, and Algorithms 1 and 2 must be run again.
ii. The node is automatically removed from the cluster's membership roster if its CM leaves.

4.3. Routing Mechanism

Both intra-cluster and inter-cluster communiqué are taken into account when solving the UAV cluster communiqué relay routing issue. Check the neighbor table first for intra-cluster communication. To improve communication efficiency and decrease communication cost, UAV nodes should: (1) node if it is in its neighbor table; (2) forward node if it is not in its neighbor table; and (3) select the next hop node for inter-cluster communication based on a weighting node. The equation (35) that describes the function used to select the inter-cluster communication channel is as follows:

$$path_{ij} = \frac{E_j}{Dis(i,j)} \quad (35)$$

E_j symbolizes the remaining energy of node j; $Dis(i,j)$ represents the distance among nodes i and j. The uppermost remaining is rather painstaking for next-hop routing.

5. Experimental Results and Investigation

Table 1 displays the various parameters used in the simulation. The outcomes from 10 separate simulated runs have been averaged for each set.

Table 1: Parameters and values

Values	Parameters
−90 dBm	Receiver Sensitivity
1	W1 +W2 + W3

1000 × 1000 m², 2000 × 2000 m² and 3000 × 3000 m²	Grid Size
20, 30, 40, 50, 60	Density of Connected Nodes
5 m	Minimum Distance Among Nodes
Reference Point Mobility Perfect	Mobility Perfect
2 s	Position Exchange Interval
80-Watt Hour	Node Energy Level at Start Period
Dynamic	Transmission Variety
2.45 GHz	Transmission Frequency
100 kbps	Constant Bit Rate
10	Simulation Runs
120 s	Simulation Time

5.1. Average Packet Delivery Ratio (PDR)

The efficiency of a network can be gauged by looking at the ratio, which is the sum of packets conventional at the terminal node compared to the total sum of packets transmitted.

$$PDR = \frac{\sum_{u=1}^{K} \frac{P_{ru}}{P_{lu}}}{K} \quad (36)$$

where P_{ru}, P_{lu}, the numbers and k in equation (36) represent the entire sum of simulation runs, the number of packets produced by the basis node during the u[th] imitation run, and the number of packets conventional by the superficial sink during the u[th] imitation run, respectively.

Table 2: PDR Investigation

Models	100	200	300	400	500	600	700
Proposed	0.22	0.53	0.72	0.85	0.91	0.92	0.92
Binary whale optimisation	0.1	0.21	0.31	0.38	0.44	0.45	0.45
Grasshopper	0.12	0.29	0.39	0.42	0.52	0.52	0.52
Weed optimizer	0.16	0.34	0.44	0.51	0.66	0.73	0.73

Table 2 characterise that the PDR Analysis. In the analysis of Binary Whale Optimisation model attained the PDR rate in 100 nodes as 0.1 and 200 nodes of 0.21and 300 node the PDR rate as 0.31 and 400 nodes as 0.38 and 500 node the PDR rate as 0.44 and 600 node the PDR rate as 0.45 and 700 node the PDR rate as 0.45 correspondingly. Then the Grasshopper model attained the PDR rate in 100 nodes as 0.12 and 200 nodes of 0.29 and 300 nodes as 0.39 and 500 node the PDR rate as 0.42, 600 nodes the PDR rate as 0.52 and 700 node the PDR rate as 0.52 correspondingly. Then the Weed optimizer model attained the PDR rate in 100 nodes as 0.16 and 200 nodes of 0.34 and 300 nodes as 0.44 and 400 node the PDR rate as 0.51 and 500 node the PDR rate as 0.66 and 600 node the PDR rate as 0.73 and 700 node the PDR rate as 0.73 correspondingly. Then the Proposed model attained the PDR rate in 100 nodes as 0.22 and 200 nodes of 0.53 and 300 nodes as 0.72 and 400 node the PDR rate as 0.85 and 500 node the PDR rate as 0.91and 600 node the PDR rate as 0.92 and 700 node the PDR rate as 0.92 correspondingly. Figure 4 represents investigation of PDR rate in graphical format.

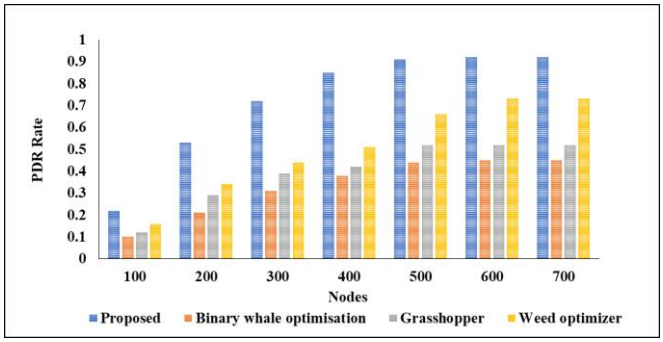

Fig 4. Graphical representation of PDR investigation.

5.2. Average End-to-End Delay(EDD)

End-to-end latency is the period it takes a packet to travel from its basis node to its final sink on the network's top layer. The -trip delay is expressed as an equation (37):

$$EED = \frac{\sum_{u=1}^{K} \sum_{m=1}^{Pr} \left\{ (TP_{um} - RP_{um}) + T_{H_{r_j}} \right\}}{P_r K} \quad (37)$$

where P_r, RP_{um}, and TP_{um} represents the total elapsed time between the sending and receiving of the m[th] packet in mock run's reception.

Table 3: End to End delay (s)

Models \ Nodes	100	200	300	400	500	600	700
Weed optimizer	68	52	46	46	44	42	41
Binary whale optimization	93	81	76	75	72	72	72
Grasshopper	84	74	69	64	57	57	56
Proposed	58	43	38	36	34	34	34

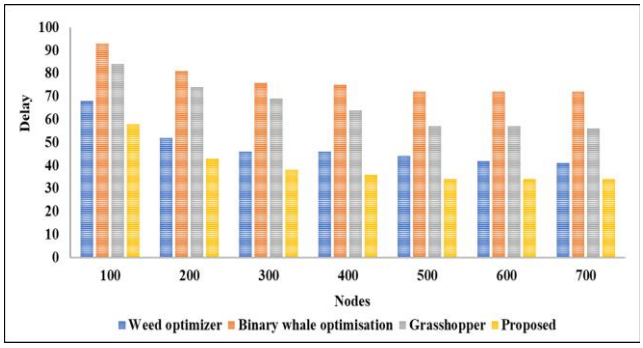

Fig 5. Graphical representation of average End-to-End Delay.

Table 3 characterizes the End-to-End delay (s). in the analysis of Binary whale optimization model reached the delay at 100 nodes as 93 and 200 nodes as 81 and 300 nodes as 76 and 400 nodes as 75 and 500 nodes as 72 and 600 nodes as 72 and 700 nodes as 72 correspondingly. Then the Grasshopper model reached the delay at 100 nodes as 84 and 200 nodes as 74 and 300 nodes as 69 and 400 nodes as 64 and 500 nodes as 57 and 600 nodes as 57 and 700 nodes as 56 correspondingly. Then the Weed optimizer model reached the delay at 100 nodes as 68 and 200 nodes as 52 and 300 nodes as 46 and 500 nodes as 46 and 500 nodes as 44 and 500 nodes as 42 and 500 nodes as 41 correspondingly. Then the Proposed model reached the delay at 100 nodes as 58 and 200 nodes as 43 and 300 nodes as 38 and 400 nodes as 36 and 500

nodes as 34 and 600 nodes as 34 and 700 nodes as 34 correspondingly. Figure 5 depicts delay period with nodes of comparison between existing and proposed models in graphical format.

5.3. Average Energy consumption

The packets' transmission, reception, and eavesdropping by the nodes in the persistence relay set all contribute to the regular amount of energy consumed for a successful transfer. Equation (38) can be used to calculate the typical network power consumption,

$$E_{Avg} = \frac{\sum_{u=1}^{K}\{F(i), r_j\}}{K} \quad (38)$$

Table 4: Validation of Projected perfect in terms of Energy Consumption (J)

Models \ Nodes	100	200	300	400	500	600	700
Proposed	32	86	120	153	190	230	260
Binary whale optimization	45	110	125	160	210	240	290
Grasshopper	55	125	135	185	245	265	330
Weed optimizer	86	140	156	210	260	290	340

Table 4 characterizes the Authentication of Projected perfect in terms of Energy Ingesting (J). in the analysis of Proposed model reached Energy Consumption in 100 nodes as 32 and 200 nodes as 86 and 300 nodes as 120 and 400 nodes as 153 and 500 nodes as 190 and 600 nodes as 230 and 700 nodes as 260 correspondingly. Then the Binary whale optimization model reached Energy Consumption in 100 nodes as 45 then 110 and 300 nodes as 125 and 400 nodes as 160 and 500 nodes as 210 and 600 nodes as 240 and 700 nodes as 290 correspondingly.

Then the Grasshopper model reached Energy Consumption in 100 nodes as 55 and 200 nodes as 125 ,300 nodes as 135 and 400 nodes as 185 and 600 nodes as 265 and 700 nodes as 330 correspondingly. Then the Weed optimizer model reached Energy Consumption in 100 nodes as 86 and 200 nodes as 140 and 300 nodes as 156 and 400 nodes as 210 and 500 nodes as 260 and 600 nodes as 290 and 700 nodes as 340 correspondingly.

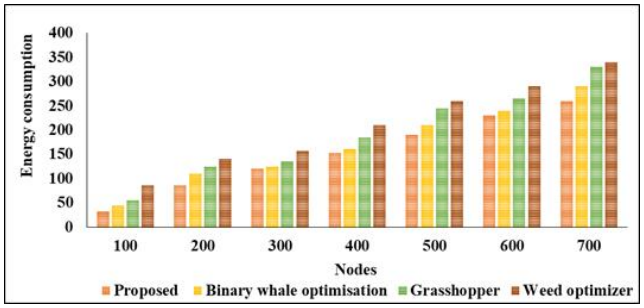

Fig 6. Graphical representation of average energy consumption.

5.4. Average network lifetime

At least until then, the network would operate normally. The network lifetime is determined as the time when the network stops receiving power. Because of this, a network's lifetime was estimated by tracking node to die over the course of the simulation. The formula for calculating the average lifespan of a network is given in equation (39), which reads as:

$$L_{Avg} = \frac{\sum_{u=1}^{K}(ST_u - FT_u)}{K} \quad (39)$$

where ST_u and FT_u occur when the primary node in the u[th] simulated run started consuming power and the simulation itself started.

Table 5: Investigation based on Network Lifetime (s)

Models \ Nodes	100	200	300	400	500	600	700
Weed optimizer	1600	1400	1200	1000	900	900	900
Binary whale optimization	1000	800	750	700	700	700	700
Grasshopper	1200	940	820	750	750	750	750
Proposed	1800	1500	1300	1150	1050	1050	1050

Table 5 characterizes the Investigation based on Network Generation. In the analysis of Binary whale optimization model reached Network Lifetime of 100 nodes as 1000,200 nodes as 800, 300 nodes as 750, 400 to700 nodes as 700 correspondingly. Then the Grasshopper model reached Network Lifetime of 100 nodes as 1200 ,200 nodes as 940,300 nodes as 820,400 to 700 nodes as 750 correspondingly. The Weed optimizer model reached Network Lifetime for 100 nodes as 1600, 200 nodes as 1400, 300 nodes as 1200, 400 nodes as 1000 and 500 nodes as 900,600 and 700 nodes as 900 correspondingly. Then the Proposed 100 nodes as 1800, 300 the 1300 and 400 nodes as 1150 and 500 nodes as 1050 and 600 nodes as 1050 and 700 nodes as 1050 correspondingly. Figure 7 provides the network lifetime comparison of existing models with proposed model in graphical format.

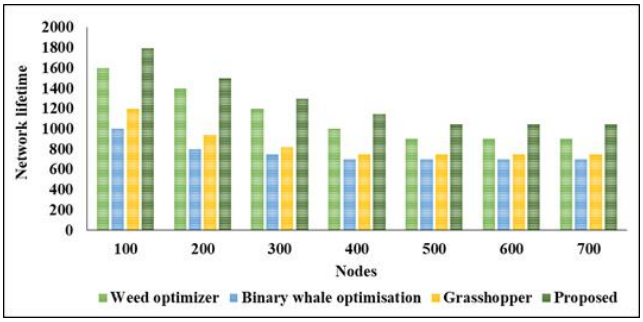

Fig 7. Graphical representation based on Network Lifetime.

6.Conclusions

Creative routing techniques are required to meet the challenges presented by vast FANETs, particularly because unmanned aerial aircraft are involved. In order to handle these intricate routing problems, clustering techniques especially those that make use of AI techniques hold considerable promise. In order to improve routing efficiency, this work presents the HIWIGOA-R technique, which combines the advantages of the GOA and the IWOA. A more balanced and successful strategy is produced by adding a random walk tactic and adjusting the exploitation coefficient through grouping. According to experimental findings, HIWIGOA-R performs better than the state-of-the-art techniques in terms of packet delivery ratio, energy consumption, end-to-end delay, and network lifetime, demonstrating its potential as a useful remedy. In order to excellent the CH with the best presentation, network's energy consumption is reduced, and the system's existence period is extended, hybrid bionic optimization algorithms may be considered in the future.

References:

[1] Wheeb, A. H., Nordin, R., Samah, A. A., Alsharif, M. H., & Khan, M. A. (2021). Topology-based routing protocols and mobility models for flying ad hoc networks: A contemporary review and future research directions. Drones, 6(1), 9.

[2] Pasandideh, F., da Costa, J. P. J., Kunst, R., Islam, N., Hardjawana, W., & Pignaton de Freitas, E. (2022). A review of flying ad hoc networks: Key characteristics, applications, and wireless technologies. Remote Sensing, 14(18), 4459.

[3] Bharany, S., Sharma, S., Badotra, S., Khalaf, O. I., Alotaibi, Y., Alghamdi, S., & Alassery, F. (2021). Energy-efficient clustering scheme for flying ad-hoc networks using an optimized LEACH protocol. Energies, 14(19), 6016.

[4] Oubbati, O. S., Atiquzzaman, M., Lorenz, P., Tareque, M. H., & Hossain, M. S. (2019). Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives. IEEE Access, 7, 81057-81105.

[5] Khan, I. U., Qureshi, I. M., Aziz, M. A., Cheema, T. A., & Shah, S. B. H. (2020). Smart IoT control-based nature inspired energy efficient routing protocol for flying ad hoc networks (FANET). IEEE Access, 8, 56371-56378.

[6] Arafat, M. Y., & Moh, S. (2021). A Q-learning-based topology-aware routing protocol for flying ad hoc networks. IEEE Internet of Things Journal, 9(3), 1985-2000.

[7] Liu, J., Wang, Q., He, C., Jaffrès-Runser, K., Xu, Y., Li, Z., & Xu, Y. (2020). QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks. Computer Communications, 150, 304-316.

[8] Kaur, M., & Verma, S. (2020). Flying ad-hoc network (FANET): challenges and routing protocols. Journal of Computational and Theoretical Nanoscience, 17(6), 2575-2581.

[9] Azevedo, M. I. B., Coutinho, C., Toda, E. M., Carvalho, T. C., & Jailton, J. (2020). Wireless communications challenges to flying ad hoc networks (FANET). Mobile Computing, 3.

[10] Agrawal, J., & Kapoor, M. (2021). A comparative study on geographic-based routing algorithms for flying ad-hoc networks. Concurrency and Computation: Practice and Experience, 33(16), e6253.

[11] Tsao, K. Y., Girdler, T., & Vassilakis, V. G. (2022). A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks. Ad Hoc Networks, 133, 102894.

[12] Arafat, M. Y., Poudel, S., & Moh, S. (2020). Medium access control protocols for flying ad hoc networks: A review. IEEE Sensors Journal, 21(4), 4097-4121.

[13] Lee, S. W., Ali, S., Yousefpoor, M. S., Yousefpoor, E., Lalbakhsh, P., Javaheri, D., ... & Hosseinzadeh, M. (2021). An energy-aware and predictive fuzzy logic-based routing scheme in Flying Ad Hoc Networks (FANETs). IEEE Access, 9, 129977-130005.

[14] Xue, Q., Yang, Y., Yang, J., Tan, X., Sun, J., Li, G., & Chen, Y. (2023). QEHLR: A Q-Learning Empowered Highly Dynamic and Latency-Aware Routing Algorithm for Flying Ad-Hoc Networks. Drones, 7(7), 459.

[15] Anwekar, D., & Phulre, S. (2023, July). Analysis of Congestion Control Techniques to Improve QoS and Frequent Communication in FANET. In 2023 World Conference on Communication & Computing (WCONF) (pp. 1-7). IEEE.

[16] . Rahman, K., Aziz, M. A., Usman, N., Kiren, T., Cheema, T. A., Shoukat, H., ... & Sajid, A. (2023). Cognitive Lightweight Logistic Regression-Based IDS for IoT-Enabled FANET to Detect Cyberattacks. Mobile Information Systems, 2023.

[17] Hosseinzadeh, M., Mohammed, A. H., Alenizi, F. A., Malik, M. H., Yousefpoor, E., Yousefpoor, M. S., ... & Tightiz, L. (2023). A novel fuzzy trust-based secure routing scheme in flying ad hoc networks. Vehicular Communications, 44, 100665.

[18] Kumar, S., Rathore, N. K., Prajapati, M., & Sharma, S. K. (2023). SF-GoeR: an emergency information dissemination routing in flying Ad-hoc network to support healthcare monitoring. Journal of Ambient Intelligence and Humanized Computing, 14(7), 9343-9353.

[19] Kumar, S., Raw, R. S., & Bansal, A. (2023). LoCaL: Link-optimized cone-assisted location routing in flying ad hoc networks. International Journal of Communication Systems, 36(2), e5375.

[20] Lansky, J., Rahmani, A. M., Malik, M. H., Yousefpoor, E., Yousefpoor, M. S., Khan, M. U., & Hosseinzadeh, M. (2023). An energy-aware routing method using firefly algorithm for flying ad hoc networks. Scientific Reports, 13(1), 1323.