A Novel Method for Node Connectivity with Adaptive Dragonfly Algorithm and Graph-Based m-Connection Establishment in MANET

S. B. Manoojkumaar[1, *] and C. Poongodi[2]

Abstract: Maximizing network lifetime is measured as the primary issue in Mobile Ad-hoc Networks (MANETs). In geographically routing based models, packet transmission seems to be more appropriate in dense circumstances. The involvement of the Heuristic model directly is not appropriate to offer an effectual solution as it becomes NP-hard issues; therefore investigators concentrate on using Meta-heuristic approaches. Dragonfly Optimization (DFO) is an effective meta-heuristic approach to resolve these problems by providing optimal solutions. Moreover, Meta-heuristic approaches (DFO) turn to be slower in convergence problems and need proper computational time while expanding network size. Thus, DFO is adaptively improved as Adaptive Dragonfly Optimization (ADFO) to fit this model and re-formulated using graph-based m-connection establishment (G-mCE) to overcome computational time and DFO's convergence based problems, considerably enhancing DFO performance. In (G-mCE), Connectivity Zone (CZ) is chosen among source to destination in which optimality should be under those connected regions and ADFO is used for effective route establishment in CZ indeed of complete networking model. To measure complementary features of ADFO and (G-mCE), hybridization of DFO-(G-mCE) is anticipated over dense circumstances with reduced energy consumption and delay to enhance network lifetime. The simulation was performed in MATLAB environment.

Keywords: Routing, connectivity zone, ADFO, mobile ad-hoc network, graph-based m-connection establishment.

1 Introduction

In networking, infra-structure models are constructed by diverse independent mobile nodes in MANETs [Jabbar (2016)]. Networks are involved in fields like disaster recovery systems, military applications, maritime communications, forestry, vehicular networks and so on, where restricted infrastructure is needed [Bratton (2007)].

[1] Department of Computer Science and Engineering, J. K. K. Munirajah College of Technology, Gobichettipalayam, Erode, 638506, India.

[2] Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam, Erode, 638401, India.

[*] Corresponding Author: S. B. Manoojkumaar. Email: manoojkumaarsb@gmail.com.

Received: 04 April 2020; Accepted: 16 June 2020.

Unpredictable and varying network topology is cause for random node mobility in MANET. Owing to inherent features, offering assistance to diverse multi-media applications is a confronting task [Priyadharshini (2016)]. In the academic-based investigation, MANET performance features like reliability, inter-networking, quality of service, security, multi-casting and power consumption that are associated with performance that has acquired more focus [Basurra (2015)]. MANET security is an appropriate element for preliminary network functionalities like routing and packet forwarding [Kumar (2016)]. With appropriate security models, it is probable for wireless networking to sense network traffic, controlled packet headers, replay transmissions and routing message re-direction, thereby network attacks are diminished [Chang and Hsiao (2016)]. Owing to MANET characteristics and vulnerabilities, there is some essential security solution that provides services like authentication, availability, confidentiality, integrity and non-repudiation [Kennedy (2011)]. MANET protection is partitioned into two types: data transmission protection and routing functionality based protection [Vallikannu and George (2015)]. Security-based routing strategies offer authentication functionalities that protect replaying and modification of route control messages and utilizes diverse cryptographic factors for offering protective routing [Cadger (2013)]. Routing protocol depicts how communication with one another and disperse information is established that facilitates it to choose routes among any two nodes in computer networking [Patel (2014)]. The ultimate objective of the routing protocol is to choose optimal path among nodes for transmission of data [Kumar and Mehfuz (2016)]. In table-driven or pro-active protocols like optimized link-state routing protocol (OLSR) and destination sequenced distance vector, every network node comprises a table that lists all probability way to reach destination and transmission routes, while in reactive protocols like AODV and DSR, routes will be established only on-demand basis [Lu and Zhu (2010)]. So, to fulfill the essential characteristics of the routing protocol, there is the necessity of certain security factors [Zhang, Zhang and Gu (2017)]. There exist two attack types: internal and external attacks [Suraj and Tapaswi (2016)].

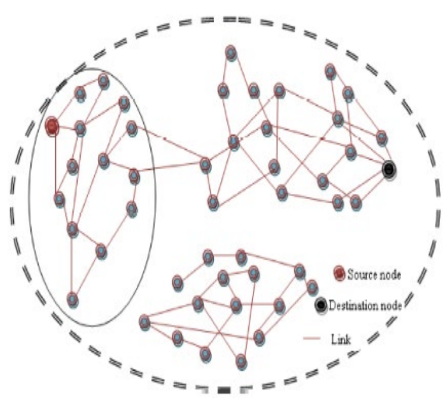

Figure 1: MANET system model

Construction of multi-cast routing protocols is a complex crisis in group membership modification, node-link fails, and nodes failure [Mandhare and Thool (2016)].

Anoptimization is an approach that provides solution to crisis relating to intense values of diverse objectives. Optimization crises are encompassed of diverse goals optimized over the same time [Moussaoui and Semchedine (2014)]. Henceforth, optimization is carried out based on multiple objective functionalities. Multi-objective crisis needs certain conditions to be fulfilled. Optimization algorithms are measured with Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Ant Colony Optimization (ACO). Even though the above-mentioned approaches are resourcefully applied to various engineering crises, alike other meta-heuristic approaches there are certain disadvantages. For instance, exploitation and exploration stages are not balanced effectually [Wang and Chen (2014)]. Diverse exploration methods can eliminate local optimization to certain degree, however, precision is not improved; whilst numerous approaches lead the algorithm to be converged in a local optimum. Therefore, numerous investigators have anticipated diverse versions of the Dragonfly Algorithm (DA) to eradicate these de-merits. Dragonfly optimization is used for solving real world problems. In DA, to investigate the search space high alignment, lower cohesion weights are used, similarly, to analyze search space low alignment, high cohesion weights are used. As well, to project exploitation and exploration, neighborhood radii are proportionally enlarged to number of iteration used. In case of Adaptive dragonfly optimization, dragonflies may adaptively change their weight. It may fulfill the individual convergence during optimization. When optimization continuous, neighborhood regions are expanded to adjust flying path. Therefore, during final stage of optimization, swarm turns to be one group to provide global optimum. The worst and best solutions are determined by food sources and enemies respectively. This work anticipates multi-objective security-based routing strategies that use hybridization with this optimization algorithm. This algorithm is cast-off for secured routing with Adaptive Dragonfly optimization and graph-based $m-$ connection establishment (G-mCE) to offer optimality in a solution using a programming model. Therefore, the anticipated ADFO-G-mCE algorithm offers optimal routes for data transmission with the recommended energy, packet delivery ratio, delay, and throughput. This work is organized as Section 2 offers routing approaches cast-off in the literature. Section 3 depicts a system model; Section 4 depicts the anticipated ADFO-G-mCE algorithm for routing Section 5 presents numerical outcomes and related discussions. Section 6 offers a conclusion and methods for future enhancement.

2 Related works

This section discusses algorithms and techniques utilized in prevailing approaches based on MANET routing without or with security, which encourages execution of anticipated security-based routing model [Moussaoui and Semchedine (2014)] depicted an approach to generate an everlasting path and generate a stable path among each MANET node pairs. This approach cast-off stability function based on mobility measure of nodes' degree and corresponding neighbor, as a selection criterion of the primary path. Functionality was utilized over OLSR to determine steady MPRs topology [He, Xie, Xu et al. (2019); Wang and Chen (2014)] have depicted QoS metric measure and trusted authority to measure TQR. Rajan [Rajan (2015)] modeled a hybrid genetic dependent optimization approach for multicast routing. It is beneficial over both features of PSO and GA for superior outcomes with utilization of Roulette wheel assortment approach to choose initial solutions. Moreover, the consequences of node pause influence routing. Persis [Persis (2015)] resolved multi-objective, unicast route optimization crisis in MANET utilizing numerous performance metrics like reliability, delay, cost, hop distance and load. Ali et al. [Ali and Shahzad (2012)] constructed a multi-objective, PSO algorithm for resolving routing constraints. This procedure optimizes the number of clusters in MANET networking and nodes energy dissipation to generate efficient solutions concerning traffic. The foremost significance of this model is to produce a solution set concurrently. Balachandra [Balachandra (2014)] had offered multi-constrained protocol and QoS aware routing protocol along with path recoveryto utilize reliable communication and energy-efficient communication paths. Gurung et al. [Gurung and Chauhan (2018)] anticipated strategy termed as Modified Genetic Algorithm with Micro-Movement (MGAM). It utilized definite nodes termed as grey hole-intrusion detection schemes for MANET nodes, therefore to prevent and identify a smarter attack. Borkar et al. [Borkar and Mahajan (2017)] had enlarged Ad hoc On Demand Multipath Distance Vector (AOMDV) protocol as the baseline protocol. Mesh-based multi-path routing approach has demonstrated probable routes that are utilizing adjacent trust verification protocol determining path using echolocation procedure for an effectual routing. Mirjalili et al. [Mirjalili and Lewis (2016)] have initiated Whale Optimization Algorithm (WOA) which is a Meta-heuristic optimization procedure inspired by nature and functions like whales to determine the finest solution. Rajakumar [Rajakumar (2014)] has initiated Lion algorithm (LA) to resolve the benchmark minimization crisis.

3 Problem description

Consider, graph-based network connectivity $G\,(W,E)$ specified as MANET node establishment where $N=\{n_1,n_2,\dots,n_n\}$ specifies set of nodes, where $'n'$ total amount of nodes in MANET networking and $'E'$ is set of edges connects two nodes $'n_x'$ and $'n_y'$ correspondingly. Nodes are recommended inside coverage region $'C_R'$ of the network. Every network edge is connected with a delay while forwarding data packets and distance among two nodes $'n_x'$ and $'n_y'$. Consider, $'S'$ as a source that broadcasts packets to destination represents $'D'$, via intermediate node $'I'$. As transmission commences, nodes energy may slowly be depleted and node turns to be dead node while energy is dissipated completely. Therefore, energy is a major constraint recommended in routing MANET Jabbar [Jabbar (2016)]. Fig. 1 depicts a MANET system model that demonstrates data transmission through the source to the destination node.

4 Proposed methodology

4.1 Energy model

Energy model presented here considers MANET with $'n'$ nodes possessing divergent transmission range which is distributed with the Euclidean plane. It is represented as $G\,(W,E)$, where $w_i\in W$ specifies node/nodes weight, where $e_i\in E$ is edged/link among nodes. The maximum and minimum nodes coverage region is specified as C_{R1} and C_{R2}. Therefore, by considering location nodes, transmission coverage is given as $C_{R1}\le C_R\le C_{R2}$. If (i_x,j_x) and (i_y,j_y), then Euclidean distance r_{xy} among nodes is formulated as $\sqrt{(i_x, j_x)^2 - (i_y, j_y)^2}$. Communication establishment among nodes $'n_x'$and $'n_y'$ prevails iff $r_{xy} \leq C_R$. Nodes' energy consumption while packet transmission from source to destination acts as a major role in path selection strategy. Multi-path channel and free space path loss model are used based on the distance among $'S'$ and $'D'$ correspondingly. If distance $'d'$ is lesser than pre-defined threshold region D_{th}, free space model (D^2) is used, else multi-path modeling is used (D^4). Therefore, energy consumption while transmission E_T among nodes $'x'$ and $'y'$ is provided in Eq. (1):

$$E_T(x, y) = E_{Te}(d_b) + E_{Ta}(d_b, d) = \begin{cases} d_b E_{Te} + d_b E_{fs} D^2 \; d < D_{th} \\ d_b E_{Te} + d_b E_{fs} D^4 \; d > D_{th} \end{cases} \tag{1}$$

where E_{Te} is transmitter electronic energy for d_b data bits, E_{Ta} is transmitter amplifier energy. $d_b E_{fs} D^2$ and $d_b E_{fs} D^4$ is amplifier energy based on distance and error rate. If transmission length is 1, then complete energy $E(i, j)$ is successful for packet communication as depicted in Eq. (2):

$$E(i, j) = \sum_{i=1} 2\, E_{Te}(d_b) + E_{Ta}(d_b, d) * d_{i,j}^{\propto} * d_b \tag{2}$$

where $\propto$ is an exponent of path loss.

4.2 Mobility model

Dynamical changing characteristics of MANET topology show variation in nodes' mobility. It is depicted with changing velocity, position and nodes acceleration. This modelling is measured as a factor that evaluates routing performance. Assume nodes n_x and n_y place in (i, j) and (i^*, j^*) at time $t = 0$. Dynamic network characteristics considers node in the new location of coverage region with angular locations φ_{nx} and φ_{ny} with changing velocities. Consider (i_1, j_1) and (i_1^*, j_1^*) at locations n_x and n_y in $t = 1$. With new instances, the node consumes a newer location and consistently maintains its location during transmission. Moreover, the transmission is capable only with minimal distances which are depicted as Eq. (3):

$$d(n_x, n_{y,} 0) = \sqrt{|i - i^*|^2 + |j - j^*|^2} \tag{3}$$

where $d(n_x, n_{y,} 0)$ is distance measurement with time $t = 0$, (i, j) and (i^*, y^*) are nodes location n_x and n_y at $t = 0$.Consider, nodes velocity as v_{nx} and v_{ny} respectively and nodes distance is provided as in Eqs. (4) and (5):

$$d_{nx} = v_{nx} * t \tag{4}$$

$$d_{ny} = v_{ny} * t \tag{5}$$

Nodes position n_x and n_y at time $'t'$ is specified as in Eq. (6):

$$n_x = i + v_{nx} * t * \cos\,(\varphi_{nx}) \tag{6}$$

$$n_y = j + v_{ny} * \; t * (\varphi_{ny}) \tag{7}$$

where φ_{nx} is angle among nodes n_x to move to the next location. At certain time instant $'t'$, the distance among nodes, when it is located at $n_x(i_1, j_1)$ and $n_y(i_1^*, j_1^*)$ as in Eq. (8):

$$d(n_x, n_{y,} i =) = \sqrt{|i - i^*|^2 + |j - j^*|^2} \tag{8}$$

where (i_1, j_1) and (i_1^*, j_1^*) are next nodes location n_x and n_y respectively.

4.3 Node lifetime model

Node lifetime is determined by its connectivity time of edges. Edges make corresponding node connectivity for complete message transmission is included for forming paths. Edge that encounters dis-connectivity is measured as crucial issues of MANET. Lifetime is determined by the mobility of nodes and co-ordinates. Assume, nodes n_x and n_y is position at $(i_{nx} j_{ny})$ and $(i_{nx}^* j_{ny}^*)$ correspondingly. Node exists for lifetime enhancement is depicted as in Eq. (9):

$$E_{LT} = \frac{-(sv + dm) + \sqrt{(s^2 + m^2) r C_R}^2 - (sm - dv)^2}{(s^2 + m^2)} \tag{9}$$

where, E_{LT} is lifetime, C_R is coverage region, s is speed, v is velocity, d is the direction and m is mobility respectively.

4.4 Dragonfly optimizer

Dragonfly based optimization is inspired by dynamically changing characteristics of dragonflies. It is anticipated by Mirjalili [Mirjalili (2015)] to resolve sub-marine propelling optimization as in Fig. 2. In dynamic behaviour, dragonflies form groups for direction migration which depicts exploration in the optimization phase. The static and dynamic swarming behaviors of dragonflies are inspiration from DFO algorithm, representing exploration and exploitation phases of meta-heuristic optimization. ADFO handles multi-objective problem with reduced computational complexity based on performance metrics like end to end delay, packet delivery ratio and energy consumption. Hariharan et al. [Hariharan, Sindhu, Vikneswaran et al. (2018)]. However, five parameters are modelled to execute dragonflies' behaviours, cohesion (C), alignment (A), separation (S), food attraction (F), Enemy (E). Mathematical modelling is provided as trails.

Dragonflies' separation is specified as S_i given in Eq. (10):

$$S_i = -\sum_{j=1}^{nn} x_i - x_j \tag{10}$$

where nn is neighbourhood nodes, x_i and x_j are present dragonfly location and neighborhood nodes correspondingly. If distance among x_i and x_j is lesser than the current location, then x_j is considered as the neighborhood of x_i, where current location modifies with an increasing amount of iterations.

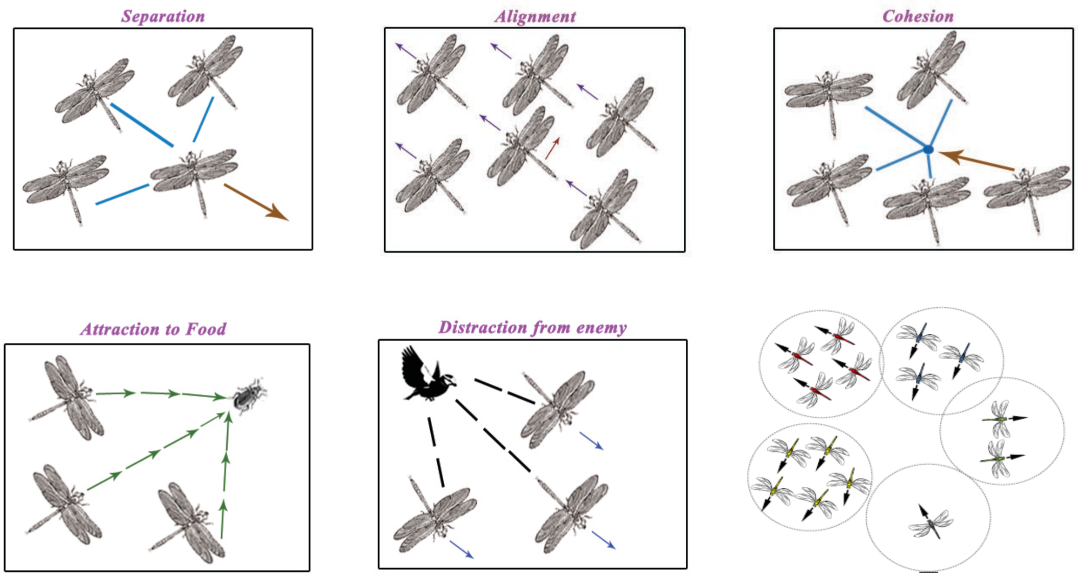

Figure 2: Original Dragonfly functions

Individual dragonfly's alignment is specified as in Eq. (11):

$$A_i = \frac{\sum_{j=1}^{nn} v_j}{nn} \tag{11}$$

where v_j is measured as dragonflies velocity. A_i is a consistent velocity measurement. Similarly, individual dragonfly cohesion is specified as C_i is evaluated as in Eq. (12):

$$C_i = \frac{\sum_{j=1}^{nn} x_j}{nn} \tag{12}$$

where x_j is dragonflies neighborhood position, food source captures the attention of n^{th} individual dragonflies as F_n and formulated as in Eq. (13):

$$F_n = x_f - x_i \tag{13}$$

where x_i is the position of dragonfly individuals. x_f is a food source position. An enemy distraction of dragonfly individual is depicted as E_d in Eq. (14):

$$E_d = x_e + x_i \tag{14}$$

where x_i is dragonfly individual and x_e is the enemy position. Specifications like x_e and x_f show the worst and best location of dragonflies searched for. The position vector of dragonflies provides individuality of intervals as $[t, t+1]$ as evaluated as in Eqs. (15) and (16):

$$x_i^{t+1} = x_i^t + \Delta x_i^{t+1} \tag{15}$$

$$x_i^{t+1} = aA_i + cC_i + eE_i + fF_i + sS_i) + \omega \Delta x_i^t \tag{16}$$

Here, Δx is an individual dragonfly directional movement, where a, c, e, f, s is alignment, cohesion, enemy, food, and separation. ω is inertial weight, t is an iterative counter.

When no neighbouring nodes are available, current dragonfly flies around for search space with random walk termed as Levy's flight to enhance algorithm performance. With the searching scenario, the dragonfly position is updated with Eqs. (17) and (18):

$$x_i^{t+1} = x_i^t + Levy'sflight * x_i^t \tag{17}$$

$$Levy'sflight = 0.01 \frac{r_1 * \delta}{|r_2|^{\frac{1}{\beta}}} * \tag{18}$$

where r_1 and r_2 are random numbers with ranges $[0,1]$, where β is constant.

4.5 Dynamic route establishment with graph connectivity for modelling adaptive dragonfly optimization (ADFO)

In a dynamic swarm, route utilization is based on allocated frequency slots. To demonstrate the physical topology of MANET, the dynamic graph is $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ with frequency slots are $\mathrm{G_{fs}} = (\mathrm{V_{fs}}, \mathrm{E_{fs}})$, where $\mathrm{v_n}$ is an adjacent node with $\mathrm{V_{i,fs}}$ and corresponding graph connectivity is $\mathrm{G_{fs}}$. With this consideration, $\mathrm{e_i}$ is node edge of $\mathrm{e_{i,fs}}$ and $\mathrm{e_i}$ is corresponding edges of $\mathrm{G_{fs}}$. With path $\mathrm{p_{fs}}$ in $\mathrm{G_{fs}}$, path constructed from primitive nodes of entire nodes on $\mathrm{p_{fs}}$and primitive edges of all paths is termed as the dynamic path. The need for constructing the lemma for ADFO is just to make an argument proof towards the nodes connectivity and data transmission in MANET. This is to prove that there is no limitation towards the connectivity establishment. Lemma 1 and Lemma 2 is different from one another as Lemma 1 concentrates on node connectivity (edges and vertices), however Lemma 2 is given for evaluating data flow among nodes. Therefore, comparative analysis cannot be provided as they may hold diverse metrics.

Lemma 1: A m −connectivity graph is $G(V, E)$ whose vertices are partitioned to source and destination sets (V_1 and V_2) where each edge is connected to vertex V_1 to V_2. Similar amount of input flow f is allocated to every source vertex of $V_1(v_i)$. The destination vertex of V_2 (v_j) is connected with weight W_i. From V_1 and V_2, every node edge is connected with transmission cost and energy cost. Unlike traditional network-basedconnectivity, the $m -$ connectivity graph has capacity limitations in every destination vertex, not on all edges. This $m -$ connectivity graph is based on the following steps:

1) All connected and failure-free nodes are measured as source and destination vertices V_1 and V_2, where v_i and v_j is related to non-connected cluster and failure-free connections correspondingly.
2) For all source node vertex v_j, the sum of input flow is a set of sensed data amounts from every cluster.
3) The capacity of every destination vertex v_j is demonstrated by accessible energy of related failure-free connectivity which is specified as in Eq. (19):

$$capacity_{v_j} = Energy\ (CH_{v_j}) \tag{19}$$

4) To illustrate edges among $v_i$and v_j, initially compute the largest probability distance among connectivity failure of CH_{v_i} and CH_{v_j} as in Eq. (20):

$$D = D_{fs}(CH_{v_i}) + D_{fs}(CH_{v_j}) \tag{20}$$

where $'D'$ is functionality for evaluating distance among two nodes. $'D'$ is to evaluate the longest distance among nodes from cluster head CH_{v_i} and CH_{v_j} that are placed in opponent directions of CH. Here, the largest probable distance can be computed. After that, $'D'$ is analyzed with a coverage region of mobile nodes. If the former is higher, it specifies connectivity terminated member of CH_{v_i} can broadcast it sensed data to CH_{v_j}. However, an edge is a connectivity among CH_{v_i} and CH_{v_j}.

5) Transmission cost of m −connectivity graph edge (v_i, v_j) is computed as in Eq. (21):

$$C = D_{CH_{v_j},sink} + D_{CH_{v_i},CH_{v_j}} \tag{21}$$

where $C(v_i, v_j)$ comprises two distance parameters. First is the distance between the sink and $CH_{vj.}$ If CH_{v_j} holds the least distance among the sink node, CH_{v_j} holds huge benefits in transmitting energy consumed by the sink node. Next is the average distance among all disconnected nodes of CH_{v_i} with CH_{v_j}. If the average distance is lesser, connectivity failed node may save energy consumption from sensing data in fault-tolerant CH.

6) Energy for fault-tolerant of m −connectivity graph (v_i, v_j) is computed as in Eq. (22):

$$E = E_{CH_i,CH_j} + E_{CH\,j} \tag{23}$$

where $E\ (v_i, v_j)$ comprises of two energy factors. The initial factor is the energy consumed by CH_j to help transmission among disconnected members of CH_i to sink node. The next factor is the energy consumed by CH_j to propagate sensed data of connected members to sink node. With provided Eq. (23) E_{CH_i,CH_j} and $E_{CH\,j}$ are formulated as in Eq. (24):

$$E_{CH_i,CH_j} = |CH_i| * D_s * E + |CH_i| * D_s * E + T(CH_j, D_s) \tag{24}$$

Based on the above computation, transmission cost based on $m -$ connectivity graph is solved with the above modeling.

Lemma 2: Consider data flow towards every connected node to establish tolerant connectivity from source to destination vertices of $m -$ connectivity graph as possible. With Equation given above, vertices and edges determine network flow connectivity. Sometimes there is some transmission failure towards destination vertex as there is a certain weighted constraint on vertex. Multi-objective cost function does not apply to Dragonfly heuristics [Rajakumar (2014)]. Thereby, Linear programming is used to acquire an optimal solution in $m -$ connectivity graph. It is measured as an optimal solution which comprises of the objective function, numerous linear constraints and integer-based solutions as in Eq. (25):

$$Min \sum_{\forall V_i \in V} \sum_{\forall\, V_j \in V} x(v_i, v_j) * t\,(v_i, v_j) \tag{25}$$

$$\forall\, v_i \in V, \sum_{\forall\, v_j \in V} x\left(v_i, v_j\right) = D \tag{26}$$

$$\forall\, V_i \in V, \forall\, v_j \in V, 0 \le x\left(v_i, v_j\right) \le D \tag{27}$$

From above Eqs. (25)-(27) shows an objective function that attempts to reduce total transmission-based connectivity. $x(v_i, v_j)$ is several connectivity-based transmission is performed among disconnected nodes in the cluster and connected nodes. Eq. (27) specifies possible values of $x(v_i, v_j)$ which ranges between 0 and D (number of sensed data), while $t\ (v_i, v_j)$ is a transmission requirement in the disconnected node. For failed CH, it is essential to sense data that has to fulfil or handle a huge amount of connected nodes. Eq. (26) is the energy constraint of connected nodes. For connected nodes in CH, it has to transmit sensed data of certain disconnected nodes along with connected nodes. Energy consumption of this connectivity cannot go beyond energy capacity as provided in Lemma 1. Linear modeling is used to acquire an optimal solution for connectivity in MANET. The flow of source vertex can be partitioned to multiple destination vertex. This shows that tolerance of disconnected CH is due to two or more connected CH. It fulfils the tolerance feature by distributing the load to connected nodes. Therefore, the algorithm functions with huge iterations. For all iterations, an edge with minimum cost is chosen with O_e. based on the chosen edge $(CH_{i,}CH_j)$ available energy E_j is measured with CH_j. Then, evaluate the energy needed for transmission by sensing data among CH_i and CH_j. After every iteration with E_i and E_j, sensed data transmission has to be updated for D_i and D_j respectively.

Algorithm 1:

Input: m −connectivity graph $G = (V, E)$ with a set of connected and disconnected nodes in CH. V_i and $V_j$are data intended to move from vertex V_i to vertex V_j inside CH.

Output: CH set of tolerant connectivity among $\left(CH_i, CH_j\right)$

1. $O_e \rightarrow$ sorting edges of $m -$ connectivity graph for establishing transmission cost
2. $CH_i \rightarrow \emptyset$
3. For $i \in V$ do
4. $D_i \rightarrow D_d$
5. End for
6. While $O_e \neq \emptyset$ do
7. Choosing cluster edge with (CH_i, CH_j) with O_e holding minimum weight
8. E_j →evaluate the energy of the present node with CH_j
9. E_i →evaluate energy needed by CH_i that has to tolerant by CH_j to broadcast D.
10. If \`$E_j \geq E_i$ then
11. $CH_t \rightarrow CH_t \cup \left(CH_i, CH_j, D_i\right)$
12. $E_j \rightarrow E_j - E_i$

13. $D_i \rightarrow 0$
14. Eliminating edges of graph connectivity with source node vertex=CH_i
15. Else
16. $D_j \rightarrow$ computing tolerant connectivity during transmission of CH_i offered by CH_j
17. $D_i \rightarrow D_i - D_j$
18. $E_j \rightarrow 0$
19. $CH_t \rightarrow CH_t \cup \left(CH_i, CH_j, D_j\right)$
20. $Edges$of disconnected nodes are removed from destination vertex= CH_j
21. $endif$
22. End while

The time complexity of the anticipated heuristic model is $O(|E|log|E$, where $|E|$ is several edges on modeling $m -$ connectivity graph.

Input: After computing Multi-objective problem with Linear programming, data flow patterns has to be examined with $\left(V_i, V_j, f\right)$

Output: Set of disconnected nodes that are connected with CH again

Sink node:

1. Delivering disconnected nodes from CH with pattern list and connected nodes.
2. Every connected CH part:
3. If it receives a tolerance message from sink nodes then
4. If CH ID fulfills pattern flow with $v_j = v_k$
5. Then
6. CH_{v_j} transmits advertisement message to all member of tolerant nodes in CH
7. $endif$
8. $endif$
9. If it receives a message from an inactive or disconnected member
10. Then
11. $CH_{v_t} \rightarrow$ retrieve ID of disconnecting nodes
12. $f_n \rightarrow (CH_{v_i}, CH_{v_j})$ to evaluate corresponding pattern (v_i, v_j, f) and retrieve
13. ifmore disconnected nodes are joined with CH
14. the
15. Send disconnection message
16. Else
17. Send acknowledgment
18. End if
19. End if

```
For all disconnected node:
If it receives the message from connected nodes in CH
Then
Maintain CH_x as tolerant connectivity
End if
CH_t maintains ID of connected nodes
While acknowledgment is not received from connected CH do
Maintaining CH from tolerant list
Transmit connectivity establishment to tolerant CH
End while
```

Here, based on connectivity establishment, CH has to handle disconnected nodes from CH. Alternatively, the message has to be transmitted to CH, as it retrieves ACK message, which exhibits fault-tolerant CH and terminates join messages.

4.6 Adaptive dragonfly optimization

Standardized DFO for solving continuous optimization crisis is not openly appropriate for Multi-objective problem [Mandhare and Thoolv (2016)]. As DFO based dynamic connectivity comes under optimization problem [Ranjini and Murugan (2017)], binary specifications are considered. The number of edges in the graph is specified as $|e|$. Fig. 3 depicts a flow diagram of ADFO.

4.6.1 Connectivity establishment mechanism

In the initial phase, for food source $|N|-1$ is randomly chosen and set as 1 and remaining nodes are determined as 0. Fitness computation is performed as in Eq. (28):

$$Fitness = \begin{cases} \sum_{e \in E_T} W_e & optimal \\ \infty & non-optimal \end{cases} \tag{28}$$

Dragonflies use this fitness computation in search of food. The position has been chosen haphazardly from source as 1 and specified as i_1. The next position is chosen randomly from Cluster as 0 and depicted as i_2. Values associated with i_1 and i_2 may be exchanged. After these modifications, sources with 1 are not intended to change. It shows that sources with 1 are constant as it is a binary string [Zhang, Zhang and Gu (2017)]. The fitness solution attained is from Eq. (28). For computing the average best position, elite strategy is used here. Sources are sorted based on fitness value from smaller to larger. For sorting sources, the mean value of food source is computed with Eq. (29):

$$number_j = \frac{\sum_{i=1}^{N/2} f_{i,j}}{N/2} \tag{29}$$

$$a = random\ (0,1) \tag{30}$$

$$p_j = \begin{cases} 1 & q_j \geq 0.5 \\ 0 & q_j \geq 0.5 \end{cases} \tag{31}$$

Here, N is the dragonfly population size, $f_{i,j}$ is component of sources based on a fitness function, i.e., initial search of a food source; $random$ (0,1) is a random decision-based return value. $'t'$ is evolutionary time for food search. p_j is sources with equal value. Position denoted as $'k'$, that is randomly chosen from $(1 - p_j)$. The position of $'j'$ and $'k'$ may be interchanged. Food source search of DF is provided as below:

Algorithm 3:

1. Choose food sources randomly based on Eq. (28) with $f_{i,j}$
2. Choose location randomly for checking food source availability
3. Consider Eq. (29)-(31) to compute position
4. If $P_j \neq F_{i,j}$ then
5. Location is chosen randomly from $(1 - p_j)$
6. Position of elements $'k'$ and $'j'$ are inter-changed
7. End if

If no finest food source for dragonflies is found, food source will not be updated. If there is no updation for a certain period, the food source has to be discarded [Lu and Zhu (2010)]. Then, the head has to randomly generate food. This process is the same for all initialization process.

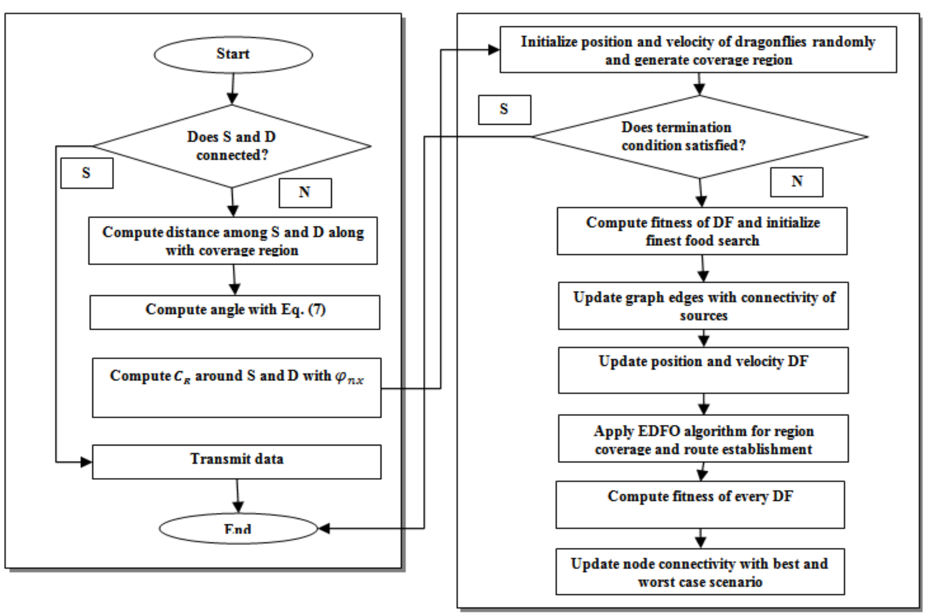

Figure 3: Flow diagram of the ADFO approach

4.6.2 Node replacement

The primary objective of node replacement based on ADFO is to fulfil food source diversity. If new food source is superior to the original food source, it is similar to other food sources. It will not be updated anymore but discarded. Initial food sources and new food searches are infeasible. However, the replacement strategy follows the replacement of the infeasible food source possibly. Algorithm 4 shows node replacement strategy:

Algorithm 4:

1. Attained solution to a new food source is not feasible, return
2. If sources are not feasible, then infeasible
3. food is randomly chosen and replace it, return.
4. Else
5. If the food source is superior to the original food source, a new food source is replaced with the original food source.

Based on this, an optimal solution can be attained with data collection and data distribution among nodes. When a connection fails, the optimality of the food source is discarded [Kumar and Mehfuz (2016)]. Initially, candidate solutions with failed connections are considered as invalid. However, the optimal solution is attained from the remaining solution.

5 Numerical results and discussions

Here, experimental evaluation of Adaptive Dragonfly optimization (ADFO) and graph-based $m-$connection establishment (G-mCE). MATLAB 2019b is used for solution Multi-objective problem optimization in MANET along with graph connectivity. In this case, PC used for configuring simulation is with 6 GB RAM and 2.2 GHz Intel Core i7 processor. Benchmark functions were considered for the optimization of Multi-objective problems. Results are evaluated and discussed with that of original routing protocols like AODV, OLSR, and DSR routing approaches with performance metrics and node connectivity [Patel (2014)].

Table 1: Simulation setup

Parameters	Values
Protocols	AODV, DSR, ADFO, OLSR
Simulation region	100 m×100 m
Number of nodes	10, 20, 30, 40, 50
Initial Energy	1 J
Transmission energy	0.01 J
Receiving energy	0.01 J
Mobility speed	1 m/s
Mobility model	Random movement
Propagation model	Propagation model
Packet size	8 bytes

There will not be any chance of problem while increasing the total amount of nodes above 50. However, the simulation setup is done with 50 nodes. In future, this can be increased for further computation. The simulation set up range is provided with 100 m×100 m for performing localization in distributed manner. The ADFO works effectually during node failure, as effectual communication is determined with connectivity. Based on Eq. (19), network has the ability to handle various nodes with reduced energy during failure condition. MANET network topology with 100 nodes depicted in Fig. 5. In Tab. 1, simulation setup is given Number of nodes considered as 10, 20, 30, 40, and 50. Based on Figs. 10 and 11, the connection establishment is done with 50 nodes, however performance metrics like Average energy consumption, packet delivery ratio and end-to-end delay is done with 30 nodes. When handling nodes that are away from sink may leads to reduce performance. Therefore, for nominal computation 30 nodes are considered.

5.1 Comparative analysis

Performance metrics of Adaptive Dragonfly optimization (ADFO) and graph-based $m-$ connection establishment (G-mCE) based routing approach is compared with prevailing approaches based on PDR, throughput, and energy. The existing models are considered for comparison which is termed as AODV, OLSR and DSR models. There is no specific reason for using the protocols for comparison. However, AODV, DSR and OLSR are considered as standard protocols in MANET. Therefore, considered these protocols with the proposed of ADFO approach. The anticipated model shows better trade-off while comparing with prevailing models like AODV [Seetaram and Kumar (2016)], OLSR [Choudhary (2015)] and DSR [Perera, Zaslavsky and Compton (2014)]. Therefore, it is considered that the anticipated model has maximal performance than previous approaches in network connectivity devoid of considering disconnectivity with all available metrics at every time instance. From the analysis carried out, it is depicted that anticipated model acquires average energy consumption; PDR and E2E delay than other models as in Fig. 4 and Fig. 7. The ultimate cause for the finest performance of the anticipated model is that fitness functionality of anticipated algorithm is designed based on factors like energy, delay, lifetime and distance [Kumar and Vidyarthi (2016); Horng (2010); Basurra (2015)]. Solutions with higher fitness values depict increment in reduced delay, energy, and higher PDR correspondingly [Kumar (2016); Chang and Hsiao (2016)].

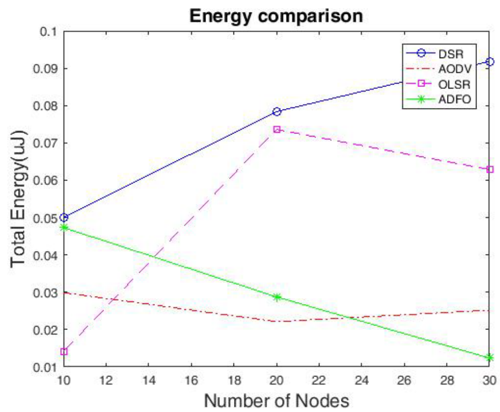

Figure 4: Nodes energy consumption

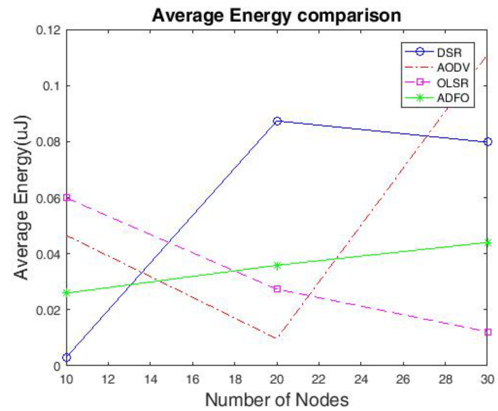

Figure 5: Average energyconsumption of nodes

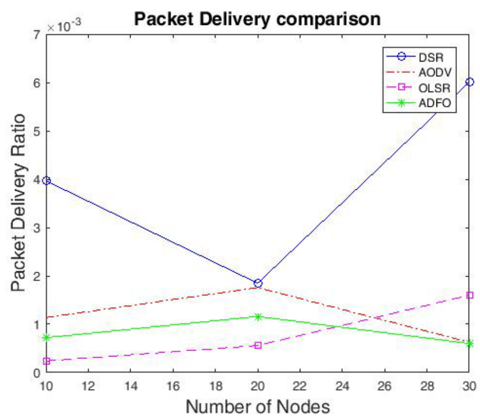

Figure 6: PDR Computation

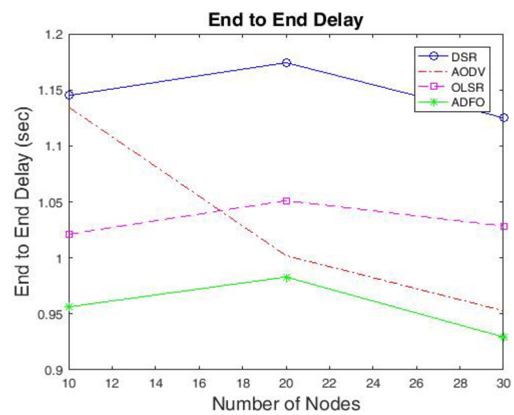

Figure 7: E2E delay computation

5.2 Multi-objective problem optimization

Here, Adaptive Dragonfly optimization (ADFO) was used for Multi-objective problems as in Fig. 8; various values are attained and are compared with other models. In ADFO, results were attained from approximately 100 iterations that will not dominate every other factor of fitness computation. While analyzing results graphically, the success of the adaptive model in terms of coverage and convergence is raised with increase in network size as in Fig. 9. While testing ADFO with graph connectivity model, search space was raised to 10 dimensionalities after performing modification. The adaptive model shows clear true success over other models. The modification in the adaptive model is compared not only with prevailing algorithms but also compared with extremely significant approaches like PSO [Jadon, Tiwari, Harish et al. (2017)], ACO [Mirjalili (2015)] with essential statistics.

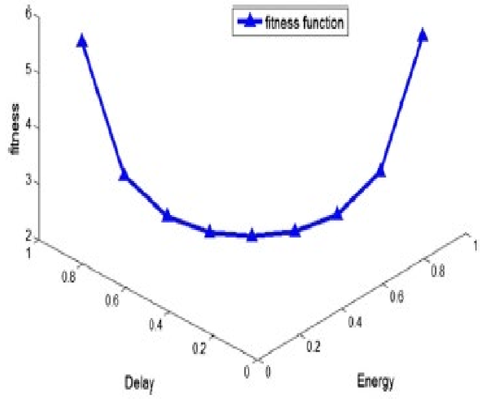

Figure 8: Fitness function

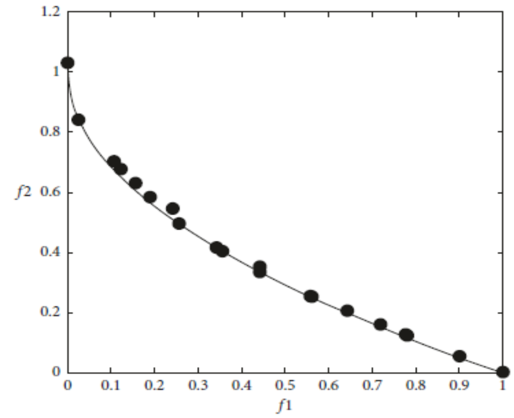

Figure 9: Convergence rate

While analyzing results and anticipated model for resolving benchmark functionalities with prior convergence problems has offered resourceful and similar outcomes with those of ACO algorithm and other models [Kennedy (2011)]. It specifies that fitness solution provided by dragonflies is similar to short step solutions of ACO, i.e., pheromone maintained by ants are nearer to others and raises its nutrient concentration that depicts its influence towards neighbourhood radius. However, PSO is measured as the most successful algorithm in SD computation [Ye and Chen (2015)]. The cause of this is conventional dragonfly algorithm has long steps that exceeds search space and rarely acquires outcomes in a shorter period. On the contrary, the anticipated model is extremely resources than PSO with local minima, i.e., short steps to the diverse direction for connectivity facilitates particles to identify various aspects and offer a superior solution at an irrelevant time interval. Adaptive DFO algorithm encounters time complexity similar to that of other algorithms. It is based on several iterations and population size. Therefore, the overall time complexity of ADFO is provided as O (iterations with maximum neighbourhood population). With graph connectivity indeed on Brownian model will not influence the time complexity of the original algorithm. The ultimate objective is to acquire optimum outcomes with the optimal amount of dragonflies. Instead of several dragonflies, the dimensionality of node connectivity and several iterations are considered as factors that influence execution time. As well, in contrary to the number of dragonflies, the concept of dimensionality with inverse proportion affects velocity. The execution time of Adaptive DFO and original DFO is also given in Tab. 3. The results of this model are compared with various dragonflies and benchmark functions.

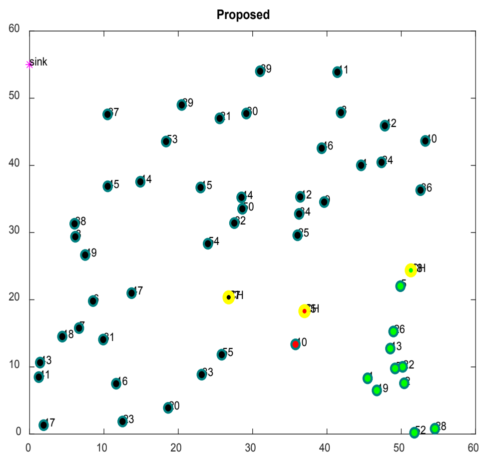

Figure 10: Coverage region

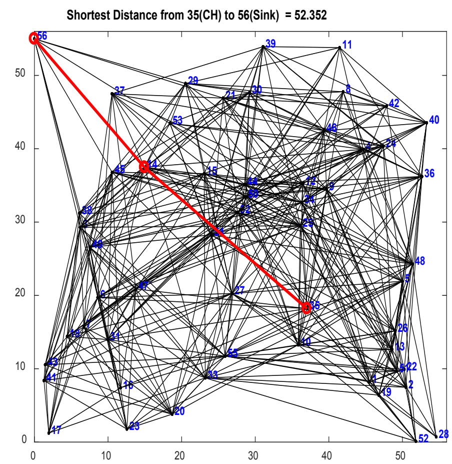

Figure 11: Connectivity with 35 CH

The time complexity of ADFO significantly depends on two factors, such as the evaluation of several neighbourhood individuals and position updates of the present dragon.

$$O(ADFO) = O(t(O(computeneighbour) + O(positionupdate)) \tag{32}$$

Table 2: Comparison of ADFO with PSO & ACO

Parameters	PSO (particles)	ACO (Ants)	ADFO (Dragonflies)
Population size	100	100	100
Iterations	150	150	150
Velocity	22-30 m/s	22-30 m/s	22-30 m/s
Acceleration	1.5 m/s^2	1.5 m/s^2	1.5 m/s^2
Maximum distance	varies	Varies	Varies
Transmission range	Dynamic	Dynamic	Dynamic
Mobility model	Freeway	Freeway	Freeway
Weight of objects	0.5	0.5	0.5

Tab. 1 determines comparative outcomes attained in network connectivity based on performance metrics at a maximum period, that is, the maximum time is 10 s. Tab. 2 depicts a comparison by considering the best outcomes attained among networks with 50-100 nodes.

Table 3: Execution time computation

Execution Time (Sec)		
Iterations	**Original DFO**	**Adaptive DFO**
1	16.050	15.250
2	15.540	16.100
3	15.640	15.740
4	15.390	15.700
5	17.120	15.425
6	14.880	15.285
7	15.780	15.390
8	17.590	15.880
9	15.010	15.650
10	15.140	14.750

ADFO and G-mCE acquire maximal performance with average energy, delay, and PDR. The proposed ADFO and G-mCE routing approach outperforms prevailing approaches with superior performance.

6 Conclusion

This work offers a programming approach for routing connectivity with Adaptive Dragonfly optimization (ADFO) and graph-based $m-$ connection establishment (G-mCE). This hybrid optimization approach combines DFO with graph theory-based modelling for optimal path selection and transmission purposes. This is a multi-objective problem that considers distance, delay, energy, lifetime and trusted connectivity. Trust is also a security factor offers security dependent trusted node degrees. As QoS and security factors are designed effectually, algorithm possesses maximal energy, lifetime, diminished delay, reduced distance, and maximal trust to construct the best path, which makes maximum fitness value. Numerous factors like total nodes, simulation region, mobility, energy, mobility, and propagation region. Adaptive Dragonfly optimization (ADFO) and graph-based $m-$connection establishment (G-mCE) performance is compared with DSR, AODV and OLSR. Metrics like throughput, energy, and PDR are essential for computation. Therefore, experimental outcomes attained with proposed model are effectual in offering routing with maximum connectivity model. The limitation associated with the proposed ADFO is complexity rises when handling large scale problems. In order to overcome this issue, hybridized algorithm model has to be constructed to deal the premature convergence to local optima. The hybridization algorithm may solve number of complex optimization problems.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Ali, H.; Waseem, S.; Khan, F. A. (2012): Energy-efficient clustering in mobile Ad-Hoc networks using multi-objective particle swarm optimization. *Applied Soft Computing*, vol. 12, pp. 1913-1928.

Balachandra, M. (2014) Multi constrained and multipath qos aware routing protocol for MANETs. *Wireless Networks*, vol. 20, pp. 2395-2408.

Basurra, S. (2015): Energy efficient zone based routing protocol for MANETs. *Ad Hoc Networks*, vol. 25, pp. 16-37.

Borkar, M.; Mahajan, A. R. (2017): A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks. *Wireless Networks*, vol. 23, pp. 2455-2472.

Bratton, D. (2007): Defining a standard for particle swarm optimization. *IEEE Swarm Intelligence Symposium*, pp. 120-127.

Cadger, F. (2013): A survey of geographical routing in wireless ad-hoc networks. *IEEE Communications Surveys & Tutorials*, vol. 15, pp. 621-653.

Chang, C. T.; Hsiao, C. Y. (2016): A location aware power saving mechanism based on quorum systems for multi-hop mobile ad hoc networks. *Ad Hoc Networks*, vol. 53, pp. 94-109.

Choudhary, A. (2015): Performance evaluation of improved reliable DSR protocol in case of node failure. *Internet Technologies and Applications*.

Gurung, S.; Chauhan, S. (2018): A novel approach for mitigating gray hole attack in MANET. *Wireless Networks*, vol. 24, pp. 565-579.

Hariharan. M.; Sindhu, R.; Vikneswaran, V.; Haniza, Y.; Thiyagar, N. et al. (2018): Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification. *Computer Methods Programs Biomed*, vol. 155, pp. 39-51.

He, S. M.; Xie, K.; Xie, K. X.; Xu, C.; Wang, J. (2019): Interference-aware multisource transmission in multi-radio and multichannel wireless network. *IEEE Systems Journal*, vol. 13, no. 3, pp. 2507-2518.

Horng, M. H. (2010): A multilevel image thresholding using the honey bee mating optimization. *Applied Mathematics and Computation*, vol. 215, pp.3302-3310.

Jabbar, W. A. (2016): Power-efficient routing schemes for MANETs: a survey and open issues. *Wireless Networks*, vol. 23, no. 6, pp. 1-36.

Jadon, S. S.; Tiwari, R.; Harish, B.; Jagdish, C. S. (2017): Hybrid artificial bee colony algorithm with differential evolution. *Applied Soft Computing*, vol. 58, pp. 11-24.

Kennedy, J. (2011): Particle swarm optimization. *Encyclopedia of Machine Learning-Springer*, pp. 760-766.

Kumar, N.; Vidyarthi, D. P. (2016): A novel hybrid PSO-GA meta-heuristic for scheduling of DAG with communication on multiprocessor systems. *Engineering Computer*, vol. 32, pp. 35-47.

Kumar, S.; Mehfuz, S. (2016): Intelligent probabilistic broadcasting in mobile ad hoc networka PSO approach. *Journal of Reliable Intelligent Environments*, vol. 2, pp. 107-115.

Kumar, V. (2016): Energy balanced position-based routing for lifetime maximization of wireless sensor networks. *Ad Hoc Networks*, vol. 52, pp. 117-129.

Lu, T.; Zhu, J. (2010): Maximizing multicast lifetime in unreliable wireless ad hoc network. *Wireless Networks*. pp. 1-11.

Mandhare, V. V.; Thool, V. R. (2016): QoS routing enhancement using metaheuristic approach in mobile ad-hoc network. *Computer Networks*, vol. 110, pp. 180-191.

Mirjalili, S. (2015): Moth-flame optimization algorithm: a novel nature inspired heuristic paradigm. *Knowledge Based System*, vol. 89, pp. 228-249.

Mirjalili, S.; Lewis, A. (2016): The whale optimization algorithm. *Advance in Engineering Software*, vol. 95, pp. 54-67.

Moussaoui, A.; Semchedine, F. (2014): A link-state QoS routing protocol based on link stability for mobile ad hoc networks. *Journal of Network and Computer Application*, vol. 39, pp. 117-125.

Patel, M. K. (2014): A hybrid ACO/PSO based algorithm for QoS multicast routing problem. *Ain Shams Engineering*, vol. 5, pp. 113-120.

Perera, C.; Zaslavsky, A.; Liu, C. H.; Compton, M. (2014): Sensor search techniques for sensing as a service architecture for the internet of things. *IEEE Sensors Journal*, vol. 14, pp. 406-420.

Persis, D. J. (2015): Ant based multi-objective routing optimization in mobile ad-hoc network. *Indian Journal of Science Technology*, vol. 8, no. 9, pp. 875-888

Priyadharshini, C. (2016): PSO based dynamic route recovery protocol for predicting route lifetime and maximizing network lifetime in MANET. *IEEE Technological Innovations in ICT for Agriculture and Rural Development*, pp. 97-104.

Rajakumar, B. R. (2014): Lion algorithm for standard and large scale bilinear system identification: a global optimization based on Lion's social behaviour. *IEEE Congress on Evolutionary Computation*.

Rajan, C. (2015): Genetic based optimization for multicast routing algorithm for MANET. *Indian Academy of Sciences*, vol. 40, no. 8, pp. 2341-2352.

Ranjini, S. K. S; Murugan, S. (2017): Memory based hybrid dragonfly algorithm for numerical optimization problems. *Expert System with Application*, vol. 83, pp. 63-78.

Seetaram, J.; Kumar, P. S. (2016): An energy aware genetic algorithm multipath distance vector protocol for efficient routing. *International Conference on Wireless Communications, Signal Processing and Networkin.*

Suraj, R.; Tapaswi, S. (2016): Mobility prediction in mobile ad hoc networks using a lightweight genetic algorithm. *Wireless Networks*, vol. 22, pp. 1797-1806.

Vallikannu, R.; George, A. (2015): Autonomous localization based energy saving mechanism in indoor MANETs using ACO. *Journal of Discrete Algorithms*, vol. 33, pp. 19-30.

Wang, B.; Chen, B. (2014): A light-weight trust-based QoS routing algorithm for ad hoc networks. *Pervasive and Mobile Computer*, pp. 164-180.

Ye, S.; Chen, B. (2015): Fuzzy entropy based optimal thresholding using bat algorithm. *Applied Soft Computing*, vol. 31, pp. 381-395.

Zhang, X.; Zhang, X; Gu, C. (2017): A micro-artificial bee colony based multicast routing in vehicular ad hoc networks. *Ad Hoc Networks*, vol. 58, pp. 213-221.