

# Routing in Wireless Mesh Networks: Three New Nature Inspired Approaches

Sharad Sharma<sup>1</sup> · Shakti Kumar<sup>2</sup> ·  
Brahmjit Singh<sup>1</sup>

Published online: 9 May 2015  
© Springer Science+Business Media New York 2015

**Abstract** Multi radio, multi hop, self organizing and self configuring wireless technology are the characteristic features of wireless mesh networks (WMNs) to offer last mile access to end users. The emergence of stochastically varying network environments critically affects routing in WMNs. Any routing policy meant for WMNs must be quickly adaptive and evolve in a decentralized self organizing and self configuring manner. This paper firstly proposes formulation of a soft computing i.e. fuzzy logic based hybrid performance metric which includes per flow (throughput, delay and jitter) as well as per node (residual energy of the node) parameters. This fuzzy logic based hybrid performance metric enumerates the integrated link cost (ILC) which is used as distance measure between two adjacent nodes. The paper further proposes three routing algorithms based upon nature inspired computing approaches namely firefly algorithm, Big Bang Big Crunch and Ant Colony Optimization. The proposed routing approaches aim at finding the minimal ILC path within a stipulated time constraint. The time constraint is governed by the mobility of network nodes. Extensive simulations were conducted for various WMN topologies. The results of the proposed approaches have been compared with two commonly used conventional approaches and were found to be far more superior. It was also observed that the self organizing capability of the proposed nature inspired routing approaches effectively reduces the complexity and makes a network quite adaptive to the dynamic network behavior found in WMNs.

---

✉ Sharad Sharma  
sharadpr123@rediffmail.com

Shakti Kumar  
shaktik@gmail.com

Brahmjit Singh  
brahmjit.s@gmail.com

<sup>1</sup> Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, India

<sup>2</sup> Baddi University of Emerging Sciences and Technology, Baddi, India

**Keywords** Wireless mesh network · Nature inspired computing · Ant Colony Optimization · Big Bang Big Crunch · Firefly algorithm

## 1 Introduction

Wireless mesh networks (WMNs) are dynamically self organizing, self configuring and easily deployable evolutionary wireless networks where nodes are capable of automatically creating and maintaining mesh connectivity between them. This kind of networking is very suitable for applications such as broadband home, community, neighborhood and enterprise networking etc. The data packets hop from one node to another until these reach the terminal node based upon multi hop transmissions. A detailed overview of WMNs discussing the challenges and open research issues of each layer, security, mobility management, cross layer design etc. is presented by Akyildiz et al. [1].

Nodes in WMNs are categorized in two classes: (1) Wireless Mesh Routers (WMRs); having either fixed location or limited mobility and (2) wireless mesh clients (WMCs). The architecture of WMNs can be classified as: (1) *Infrastructure/Backbone WMN*: In infrastructure type WMNs WMRs form an infrastructure mesh for WMCs. (2) *Client WMNs*: client WMNs provide peer-to-peer networks and WMCs perform routing and self-configuration functions too. (3) *Hybrid WMNs*: The most applicable WMN architecture-hybrid mesh is the combination of infrastructure and client meshing. WMCs can access the network through WMRs as well as directly meshing with other WMCs [1]. Popular commercial applications, architectures and key research areas are discussed by Bruno et al. [2].

WMNs are highly dynamic networks. This unpredictable dynamic network conditions critically affect the performance of WMNs. It is required that routing policies must work in a decentralized, self-organizing and self configuring way, while optimizing network resource utilization and fulfilling QoS requirements [3, 4]. The objective of routing policies is to maximize probability of data delivery, minimize delay, maximize throughput, minimize energy consumption, dynamically balancing the traffic load etc. It was shown by Decouto et al. [5] that conventional shortest path routing metric is not an affordable criterion to ascertain good paths. The performance metrics and routing issues in WMNs are discussed in detail by Waharte et al. [6]. The performance parameters in a WMN can be categorized as per flow (delay, packet loss ratio, delay jitter, hop count, throughput and interference); per node (computation complexity and power efficiency); per link (link quality, channel utilization, transmission rate, and congestion); inter flow (interference and fairness) and network wide parameters (QoS, total throughput etc.). Commonly used performance metrics are hop count, Expected Transmission Count (ETX), Expected Transmission Time (ETT), energy consumption and availability of reliable paths. Routing protocols can further be categorized based upon routing philosophy, network organization, location awareness and mobility management [6].

Due to complexities and constraints in exact reasoning based routing in dynamic wireless networks, nature inspired soft computing techniques have attracted significant attention from the research community. A detailed description of various nature inspired meta-heuristic algorithms including ACO, bee algorithm, bat algorithm, cuckoo search, firefly algorithm, particle swarm optimization (PSO) etc. is presented by Yang [7]. A genetic algorithm (GA) based dynamic shortest path routing approach was proposed by

Yang et al. [8]. Soft computing techniques i.e. GA and Neural Networks approach was applied to optimize quality of service (QoS) parameters for channel allocation in cellular networks [9]. A hybrid combination of multi objective particle swarm optimization (MOPSO) and GA is proposed by Benyamina et al. [10] to optimize the performance of WMNs. An approach for routing in Mobile Ad hoc Networks (MANETS) based upon swarm intelligence is proposed by Caro et al. [11]. This approach was inspired from Ant Colony Optimization (ACO) algorithm [12, 13]. An insect society collective behavior based routing for next generation networks was reported by Farooq et al. [14].

In this paper, we propose optimized shortest path routing strategies appropriate for dynamic WMNs. We explored ACO, BB-BC and Firefly based optimization algorithms to evaluate near shortest path under a given time constraint as dictated by the dynamics of WMN. We first create a mesh network of all those nodes which are within a specified radio range of at least one of the network nodes. We then periodically update and maintain the network by including the new nodes entering the radio range of network and deleting those nodes which have left the radio range of the network. After a network is established, all network nodes compute the integrated link cost (ILC) for the adjacent nodes. This ILC decides the path length that exists between a given source and destination (terminal) node pair. On the given information by the ILC module for adjacent nodes, optimal paths/near shortest paths are enumerated. In this paper we enumerate near shortest path using new ACO, BB-BC and Firefly based shortest path evaluation algorithms. Based upon evaluated near shortest paths, routing tables are updated periodically, thus taking care of dynamically changing environment. This ensures that most of the times routing is near shortest path based routing. The proposed algorithms were simulated in MATLAB environment and the routing performance of these proposed algorithms is compared with Adhoc On-demand Distance Vector (AODV) [15] and Dynamic Source Routing (DSR) [16] algorithms.

The remainder of the paper is organized as follows. Section 2 explains the formulation of hybrid performance metric based upon fuzzy logic. Section 3 presents the description of proposed soft computing based routing approaches. Node architecture for the proposed system model and the proposed routing algorithm are presented in Sect. 4. Section 5 presents the detailed results and discusses the performance and applicability of the proposed algorithms. Finally the conclusions are drawn in Sect. 6.

## 2 Performance Metric Formulation

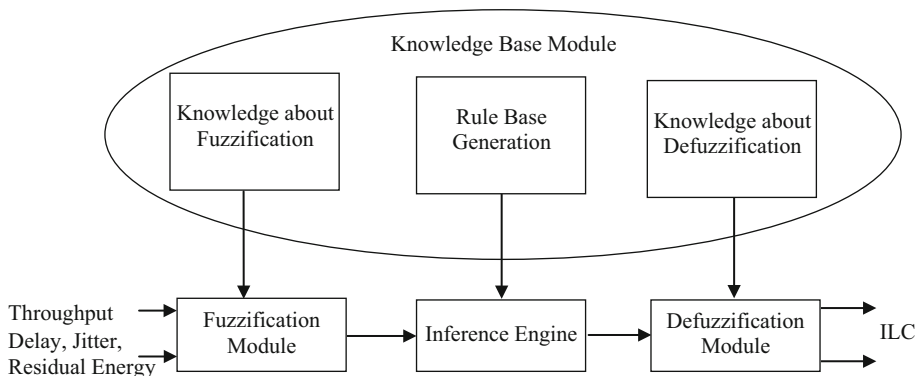
Fuzzy logic has emerged as a simple yet powerful methodology to solve non-linear functions of arbitrary complexities with vague, ambiguous or incomplete information while extracting definite conclusions [17]. We choose fuzzy logic as a suitable candidate to evaluate the integrated link cost (ILC) of WMNs based upon various input parameters. A fuzzy link cost based approach to achieve quality of service (QoS) and quality of experience (QoE) in WMNs is proposed by Gomes et al. [18]. In wireless multimedia sensor networks, fuzzy logic controller for traffic load parameter with priority based rate reduces the delay and loss probability [19]. The proposed ILC measure in this paper consists of four vital parameters of the network and nodes: throughput, end-to-end delay, jitter of the link and the residual energy of the node. For a link between adjacent nodes high throughput, low end-to-end delay and low jitter are the required conditions. A variety of applications in 'always on' dynamic multi-hop WMNs require optimal use of node energy. As WMRs deal with more heavy traffic load, optimized use of energy at WMRs is more significant.

Keeping in mind the constraints on residual energy of nodes in wireless sensor networks an intra grid architecture [20] as well as new hierarchical clustering topology architectures [21] are proposed to obtain a longer network lifetime. Thus considering the significance of residual energy of a node we have included this parameter to optimize the performance of WMNs. The node having less energy must be used accordingly for hopping or other routing purposes. Based upon these four parameters the fuzzy logic evaluated ILC measure is used as the distance between the two particular adjacent nodes.

A basic fuzzy system as shown in Fig. 1 consists of four major modules: Fuzzification module, Inference engine, knowledge base module and Defuzzification module [22]. The fuzzification module transforms the crisp input(s): throughput, delay, jitter of the link and residual energy of the node into corresponding fuzzy values (linguistic variables). These fuzzy values are then processed (rule composition, implication and aggregation) in fuzzy domain by inference engine based upon the knowledge supplied by the domain expert(s) and produces fuzzy set as output. Defuzzification module transforms the processed output of inference engine from fuzzy domain to crisp domain.

The knowledge base module contains the knowledge of the application domain and the procedural knowledge. It consists of a data base and linguistic (fuzzy) control rule base or production rules. The data base furnishes the necessary definitions which are used to define linguistic control rules and fuzzy data manipulation in an fuzzy logic Controller. The rule base characterizes the control goals and control policy of the domain expert by a set of linguistic control rules.

Firstly the current measured values of these inputs are fuzzified at fuzzification module. This module receives the crisp inputs. These input variables are then scale mapped that is transferred from the range of values of input variables into corresponding universe of discourse (membership values). The input crisp variables transformed to fuzzy variables. The universe of discourse of fuzzy variables is divided into fuzzy subsets. The input variable is mapped into these subsets with varying grade of membership that are limited to an interval between 0 and 1, i.e., any value between 0 and 1 can express the membership degree of a certain element of the fuzzy set based on the inference functions used. Usually, the relevance degree of a value ' $\chi$ ' regarding a membership function is represented by  $\mu(\chi)$ . If A is a fuzzy set in a universe U, the membership of  $\chi$  in A is evaluated by the membership function  $\mu(A)$  as following [23]:



**Fig. 1** A basic fuzzy system with input and output parameters

$$\mu_A(\chi): U \rightarrow [0, 1] \tag{1}$$

And fuzzy set A is represented as

$$A = \{i, \mu_A(i)\} \quad \forall i \in A \tag{2}$$

There are largely three types of fuzzifiers: singleton fuzzifier, Gaussian fuzzifier, and trapezoidal or triangular fuzzifier.

Figure 2 illustrates the commonly used triangular and trapezoidal membership functions. The triangular and trapezoidal membership functions are specified as follows:

$$\text{Triangle}(\chi: a, b, c) = \begin{cases} 0 & \chi \leq a \\ \frac{\chi - a}{b - a} & a \leq \chi < b \\ \frac{c - \chi}{c - b} & b \leq \chi < c \\ 0 & \chi \geq c \end{cases} \tag{3}$$

$$\text{Trapezoid}(\chi: a, b, c, d) = \begin{cases} 0 & \chi < a \\ \frac{\chi - a}{b - a} & a \leq \chi < b \\ 1 & b \leq \chi < c \\ \frac{d - \chi}{d - c} & c \leq \chi < d \\ 0 & \chi > d \end{cases} \tag{4}$$

Four linguistic input variables are Throughput, End-to-end Delay, jitter and residual energy and integrated link cost is the linguistic output variable. Low, medium and high are the linguistic values (fuzzy sets) for Throughput, jitter, Residual energy and ILC. For End-to-end Delay less, medium and high are the linguistic values. Membership functions of inputs as well as output: are shown in Fig. 3.

$$\text{Integrated Link Cost (ILC)} = f(\text{Throughput, Delay, Jitter, Residual\_Energy}) \tag{5}$$

Integrated link cost is a function of throughput, delay, jitter of the link and the residual energy of the node. Firstly the current measured values of these inputs are fuzzified and then processed by the inference engine where rule composition takes place. One such rule is given as:

If *Throughput* is **High** AND *Delay* is **Less** AND *Jitter* is **Low** AND *Residual\_Energy* is **High** then *ILC* is **Low**.

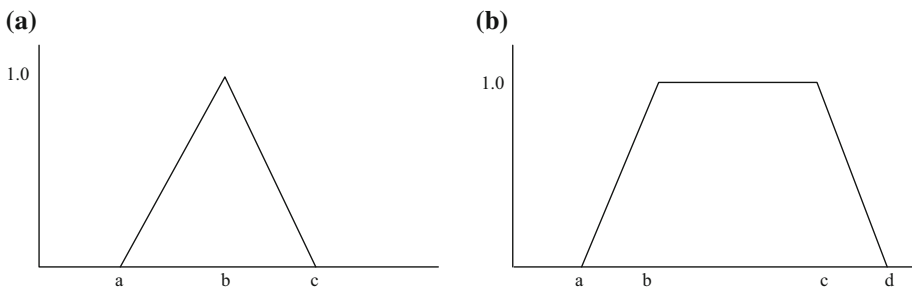
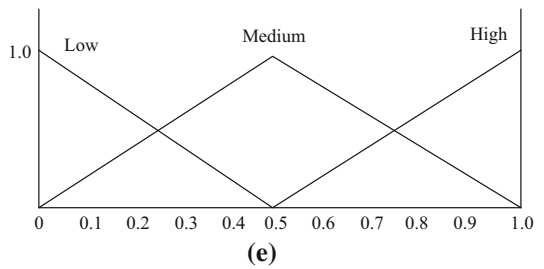
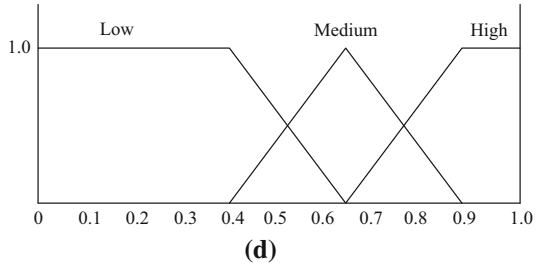
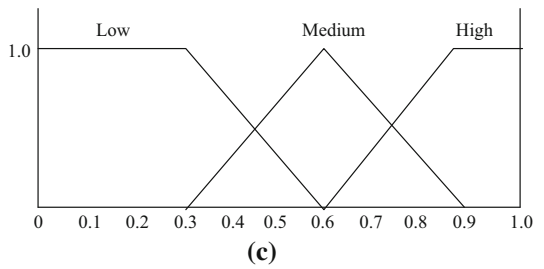
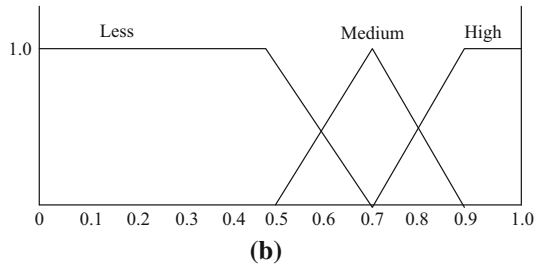
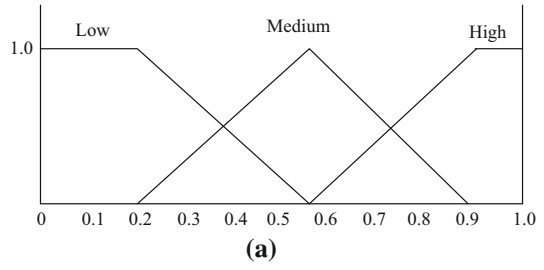


Fig. 2 Triangular and trapezoidal membership functions

**Fig. 3** Membership functions for the fuzzy set values. **a** Throughput, **b** end to end delay, **c** jitter, **d** residual energy, **e** integrated link cost



Here linguistic variables *Throughput*, *Delay*, *Jitter* and *Residual\_Energy* are the inputs; *ILC* is output; **High**, **Medium**, **Less** and **Low** are the linguistic values (fuzzy sets) in their universe of discourse. The input and output parameters with their fuzzy sets are shown in Table 1. In our proposed module  $3 \times 3 \times 3 \times 3 = 81$  rules are formed.

These rules are then implicated by Mamdani implication process [24]. The output of each implicated rule is a fuzzy set and all these fuzzy sets are then aggregated. The resulting one aggregated fuzzy set is then defuzzified to present a single crisp output (integrated link cost).

From Fig. 3e the defuzzified output value is calculated as:

$$ILC^* = \frac{\int \mu_i(x).xd(x)}{\int \mu_i(x)d(x)} \tag{6}$$

where  $\mu_i(x)$  is the membership of the output of *i*th rule (rule strength) and  $x_i$  is the weight associated with the *i*th rule (action taken).

### 3 Proposed Algorithms

#### 3.1 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is a population based meta-heuristic approach to solve combinatorial optimization problems by offering ‘good enough’ solutions in appropriate time [11, 13]. Social insect societies e.g., ant colonies are distributed systems that collectively exhibit intelligent behavior. ACO is inspired by the food searching (foraging) behavior of the real ants. Initially ants search for food randomly. After finding a food source, ants return to their nest while depositing a chemical pheromone trail on the ground. Other ants are guided to the food source by the quantity of pheromone deposited on various paths.

The Simple ACO (S-ACO) algorithm for finding the shortest path between source and terminal node in a network is as follows:

A completely connected, directed topology graph  $G(N, L)$  having  $N$  nodes and  $L$  links between adjacent nodes with a positive weight (cost)  $d_{ij}$  is given. Neighborhood of an ant  $k$  at node  $i$  is given as  $N_i^k$  while  $L^k$  is the length of ant  $k$ 's path. Minimizing the cumulative length of the path (cost) is considered as objective function. Initially a constant amount of pheromone  $\tau_{ij}$  between *i*th and *j*th node is assigned to all arcs of the graph. At a node  $i$ , the probability of selecting node  $j$  as next node with pheromone trails  $\tau_{ij}$  by ant  $k$  is computed as follows:

**Table 1** Input and output parameter membership functions

Input parameters				Output parameter
Throughput (T)	End-to-end delay (D)	Jitter (J)	Residual energy (R_E)	Link cost
High	Less	Low	High	Low
Medium	Medium	Medium	Medium	Medium
Low	High	High	Low	High

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha} & \text{if } j \in N_i^k \\ 0, & \text{if } j \notin N_i^k \end{cases} \quad (7)$$

where  $\alpha$  is a constant. Previous node is not taken as next node to avoid loops. The ant  $k$  at node  $i$  further checks its neighborhood and hops from node to node using this stochastic decision criterion until the termination node is achieved. After reaching at terminal node, the forward ant  $k$  is converted into a backward ant to retrace the travelled path. The aim of this path retracing is to update the routing information at each node while eliminating the loops. Adaptation to new routing information is regulated by  $\alpha$ . While returning to the source node, the ant  $k$  updates the pheromone value of arc  $(i, j)$  as

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau^k \quad (8)$$

where  $\Delta\tau^k$  is the increment in pheromone quantity. For shorter paths more pheromone is deposited. The deposited pheromone trails at each arc evaporates similarly to normal evaporation of real pheromone in nature. The objective of the pheromone diffusion process is to avoid quick convergence to a sub optimal route. At arc  $(i, j)$  pheromone trails are updated as

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}, \rho \in (0, 1] \quad (9)$$

where  $\rho$  is an evaporation constant. The routing tables obtained are updated at regular interval or at detection of a change in network configuration.

Pseudo Code of the ACO based proposed routing algorithm for WMNs is as given in Fig. 4.

---

```

begin
  / Initialize WMN Parameters/
  Define Source Node, Terminal Node, Number of ants, Number of Paths, Number of
  Iterations, Number and location of the nodes
  /For Static WMNs nodes are fixed while for Client and Hybrid WMNs all nodes other/
  /than Mesh Routers are having Random Locations/

  while ( $t < \text{MaxGeneration}$  or Termination Criteria not met)
    for  $i = 1 : n$  / all n Nodes /
      for  $j = 1 : n$  / all n Nodes /
        if distance  $(i, j) \leq R$  (radio range of a node)
          connectivity_matrix $(i, j) = 1$  /routing table maintenance/
          Integrated_Link_Cost  $(i, j) = f(\text{Throughput, Delay, Jitter, Residual_Energy})$ 
          /Integrated Link Cost Evaluation using Fuzzy System/
        end if
      end for  $j$ 
    end for  $i$ 
    / Build paths between source and terminal node /
    Randomly generate initial population of  $k$  paths
    Compute the shortest path using S-ACO
    Update the Pheromone Trails

  end while
  Postprocess results and visualization;
end

```

---

**Fig. 4** Pseudo Code of the ACO based routing algorithm applied in WMNs



### 3.2 Big Bang Big Crunch (BB-BC)

The Big Bang theory is one of the most widely accepted theories of the evolution of this universe [25]. It is believed that that universe began about twelve to fifteen billion years ago in inflation similar to an explosion like situation. The nature of inflation was such that for an incomprehensibly small fraction of a second, the universe was an infinitely dense and infinitely hot fireball. A peculiar form of energy that we don't know yet, suddenly pushed out the fabric of space time in a process called "inflation", which lasted for only one millionth of a second. Thereafter, the universe continued to expand but with slower pace.

The BB-BC theory believes that energy discharged by the initial explosion i.e., kinetic energy, is counterbalanced by the energy of bodies attraction known as gravitational pull. If there is enough mass so that the later is bigger than the first when a critical density is reached, the expansion will stop and the universe will start to contract, leading to an end very similar to its beginning, named by the scientists as the big crunch (great implosion). In the Big Bang phase, energy dissipation produces disorder and randomness as the main feature of this phase. In the big crunch phase, randomly distributed particles are drawn into an order. This theory of repeated big bang followed by big crunch phases forms the basis of an optimization algorithm called the Big Bang-Big Crunch optimization algorithm [26, 27].

---

```

begin
    / BB-BC Parameter Initialization/
    Define Source Node, Terminal Node, Number of Paths, Number of Iterations,
    Number and location of the nodes
    / End of BB-BC Parameter Initialization/
while (true)
    for i = 1 : n                                     / all n Nodes /
        for j = 1 : n                                 / all n Nodes /
            if distance (i, j) <= R (radio range of a node)
                connectivity_matrix(i, j) = 1          /routing table maintenance/
                Integrated_Link_Cost (i, j) = f(Throughput, Delay, Jitter, Residual_Energy)
                /Integrated Link Cost Evaluation using Fuzzy System/
            end if
        end for j
    end for i
    / Build paths between source and terminal node /
    while (t < MaxGeneration or Termination Criteria not met)
        Randomly generate initial population of k paths /Big Bang Phase/
        Compute the ILC of all the candidate solutions
        Sort the population from best to worst based on ILC /No.1 path is the Optimal path/
        Compute the center of mass  $x^c$  /Big Crunch Phase/
        Generate new candidate solutions around  $x^c$  by adding or subtracting a normal
        random Number
    end while
    wait for stipulated time/ wait for an event
end while
Postprocess results and visualization;
end

```

---

**Fig. 5** Pseudo Code of BB-BC based algorithm for optimal path evaluation in WMNs

### 3.2.1 BB-BC Optimization Algorithm

Initially a set of candidate solutions (population) is generated randomly in the search space. The fitness as defined by the objective function, of each solution is calculated and ranked accordingly. After the random Big Bang phase contraction is applied in big crunch phase to compute the centre of mass as:

$$x^C = \frac{\sum_{i=1}^N \frac{1}{f^i} x^i}{\sum_{i=1}^N \frac{1}{f^i}} \quad (10)$$

where  $x^c$  = position of the centre of mass;  $x_i$  = position of  $i$ th candidate;  $f^i$  = fitness function value of candidate  $i$ ;  $N$  = population size.

Alternatively best fit individual can also be considered as the centre of mass in place of using Eq. (5). There after we generate new population around the centre of mass by adding or subtracting a normal random number whose value decreases as the iterations elapse. This can be formalized as:

$$x^{new} = x^C + lr/k \quad (11)$$

where  $x^C$  stands for center of mass,  $l$  is the upper limit of the parameter,  $r$  is a normal random number and  $k$  is the iteration number in process. Thus new point  $x^{new}$  is upper and lower bounded. Figure 5 lists the Pseudo Code of BB-BC based Algorithm for optimal path evaluation in WMNs.

### 3.3 Firefly Algorithm (FA)

Firefly algorithm (FA) is a nature inspired evolutionary algorithm. It is inspired by the flashing light characteristics of fireflies. Fireflies produce short and rhythmic flashes to attract other fireflies, prey as well as a protective warning mechanism. The flashing light is produced by a process of bioluminescence. The flashing characteristics of fireflies can simply be described in three idealized rules [28–30]:

1. All the fireflies are unisex, so that attraction among them is unisexual.
2. Attractiveness is proportional to the brightness of the flash and the fireflies tend to move towards the brighter one in case of two flashing fireflies. Brightness decreases as their distance increases. It will move randomly, if no one is brighter than a particular firefly.
3. The brightness of a firefly is affected or determined by the landscape of the objective function.

In the case of continuous optimization problem e.g. optimization of routing path in WMN, the task is to minimize cost function  $f(x)$  for  $x \in S$  ( $S$ -sample space). Cost function ILC is integrated cost function in the case of WMNs. For firefly  $i$ ,  $x_i$  represents a solution whereas  $f(x_i)$  denotes its ILC.

Initially all fireflies are dislocated in the sample space  $S$  randomly or deterministically following some strategy. Variation of light intensity and formulation of the attractiveness are the critical issues to be considered. Simply, it is assumed that the attractiveness of a firefly is determined by its brightness or light intensity which is related with the objective function.

The brightness  $I$  of a firefly at  $x$  can be denoted as  $I(x) \propto f(x)$ . The attractiveness  $\beta$  varies with distance  $r_{ij}$  between  $i$ th and  $j$ th firefly. The attractiveness varies with the degree of

absorption as light intensity decreases with distance. The light intensity, is a function of distance  $r$ , described as

$$I(r) = I_0 e^{-\gamma r} \tag{12}$$

where  $I_0$  is the actual light intensity and  $\gamma$  is the light absorption coefficient. The attractiveness  $\beta$  of a firefly can be represented as

$$\beta = \beta_0 \exp(-\gamma r^m), \quad (m \geq 1) \tag{13}$$

where  $r$  is the distance between two fireflies and  $\beta_0$  is their attractiveness at  $r = 0$  and  $\gamma$  is a fixed light absorption coefficient. The distance between any two fireflies  $i$  and  $j$  at  $x_i$  and  $x_j$  respectively is computed by the Cartesian distance

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \tag{14}$$

where  $x_{i,k}$  is the  $k$ th component of spatial co-ordinate of  $x_i$  and  $d$  is the number of dimensions. In two dimensional case distance is

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{15}$$

The movement of lesser bright firefly  $i$  towards another brighter firefly  $j$  is evaluated as

$$\mathbf{x}_i = \mathbf{x}_i + \beta_0 e^{-\gamma r_{ij}^2} (\mathbf{x}_j - \mathbf{x}_i) + \alpha \left( \mathbf{rand} - \frac{1}{2} \right) \tag{16}$$

---

```

begin
    Define Source Node, Terminal Node, Number of Fireflies, Number of Paths, Number of
    Iterations, Number and location of the nodes
    / FA parameter initialization /
    Define Objective function  $f(\mathbf{x})$ ,  $\mathbf{x} = (x_1, \dots, x_d)^T$ ;
    Generate initial population of fireflies  $\mathbf{x}_i$  ( $i = 1, 2, \dots, n$ )
    Compute the light intensity  $I_i$  at  $\mathbf{x}_i$  determined by  $f(\mathbf{x}_i)$ ;
    Define light absorption coefficient  $\gamma$ ;
    /End of FA parameter initialization /
    while ( $t < \text{MaxGeneration}$  or Termination Criteria not met)
        for  $i = 1 : n$  / all  $n$  fireflies /
            for  $j = 1 : i$  / all  $n$  fireflies /
                Evaluate the distance  $r$  between firefly  $i$  and firefly  $j$ ;
                if ( $I_j > I_i$ )
                    Move firefly  $i$  towards  $j$  in  $d$ -dimension;
                end if
                Attractiveness varies with distance  $r$  via  $e^{-\gamma r}$ ;
                Evaluate new solutions and update light intensity;
            end for  $j$ 
        end for  $i$ 
        Rank the fireflies and find the current best;
    end while
    Postprocess results and visualization;
end
    
```

---

Fig. 6 Pseudo Code of the firefly algorithm

here the second term is due to the attraction as shown in Eq. (8) while the third term is randomization with  $\alpha \in [0, 1]$  being the randomization parameter. “rand” is a random number generator uniformly distributed in  $[0, 1]$ . Mostly  $\beta_0 = 1$  and  $\alpha \in [0, 1]$ . Pseudo Code of the firefly algorithm is shown in Fig. 6.

### 4 Node Architecture

Figure 7 shows the proposed node architecture for WMN nodes. At each node there may be multiple inputs ( $X_{i_1}, X_{i_2}, \dots X_{i_n}$  arriving from  $n$  adjacent nodes) and multiple outputs ( $Y_{o_1}, Y_{o_2}, \dots Y_{o_m}$  forwarded towards  $m$  adjacent nodes). The Node Processing Unit (NPU) makes the decision of selecting the optimal links for corresponding communication within the constraints imposed by the network dynamics and assigns time slots through a queuing system for data packets to be routed. NPU provides this link state information to Parameter Evaluation Module (PEM) to evaluate various link parameters e.g. throughput, delay, jitter and residual energy of the node. Based upon this information a fuzzy logic based system evaluates the  $ILC_i$  for the corresponding  $i$ th link. The routing tables of all nodes are updated periodically or on the occurrence of some event. This new information from the routing table is used for routing purposes.

#### 4.1 The Proposed Routing Algorithm and Validation

The proposed routing algorithm for establishing the near shortest path for a given source-terminal pair is as follows:

##### 4.1.1 Algorithm

WMN initialization phase:

1. Create a WMN by adding the nodes falling within the radio range of mesh network.

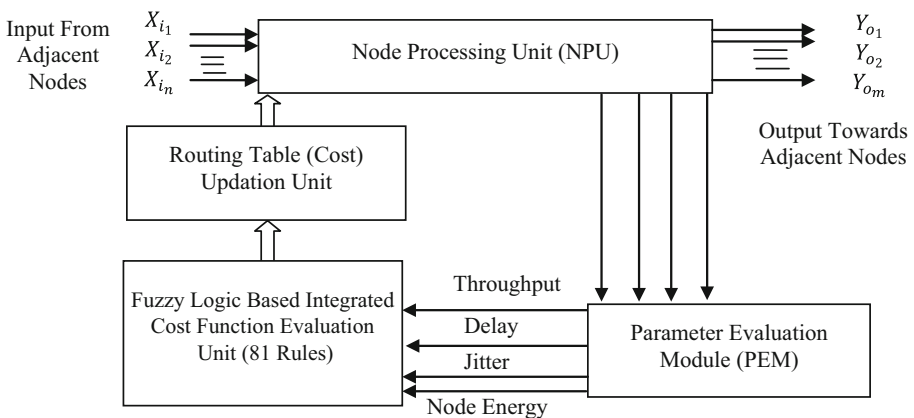


Fig. 7 Node architecture of a WMN node

2. Generate equal traffic from one node to its all adjacent nodes. Assuming equal throughput, network delays, jitter and full residual energy at each node. For each node compute shortest path to all other nodes of the network.
3. Create the routing tables and allow network traffic to flow.

WMN operational phase:

1. Maintain the network by adding incoming nodes and deleting the outgoing nodes.
2. Compute the ILC based distance, between the adjacent nodes.
3. For a given source terminal pair, evaluate the near shortest/shortest path using any of the nature inspired algorithms (ACO/BB-BC/FA).
4. Update the routing tables
5. Wait for stipulated time governed by the network dynamics.
6. Go to step 1.

### 4.1.2 Model Performance

In order to investigate and optimize the performance of routing algorithm of WMNs simulations were performed for various static and dynamic scenarios in MATLAB. We considered 9, 16, 25, 64 and 100 node networks for infrastructure WMN. These networks were placed within a 500 m × 500 m, 1000 m × 1000 m, 2000 m × 2000 m area. For client and hybrid WMNs. 10, 20, 30, 50 and 100 node networks were considered with an area of 500 m × 500 m, 1000 m × 1000 m, 2000 m × 2000 m. We varied transmission range of the nodes from 250 meters to 500 meters. In all the network models node number

**Table 2** Architectural details of WMNs

Infrastructure WMN			Client WMN			Hybrid WMN
No. of nodes	Area (m × m)	Radio range of a node (m)	No. of nodes	Area (m × m)	Radio range of a node (m)	No. of mesh routers
9	500 × 500	200	10	500 × 500	250	2
16	500 × 500	200	20	500 × 500	250	4
25	500 × 500	200	30	1000 × 1000	350	6
64	1000 × 1000	200	50	1000 × 1000	350	9
100	2000 × 2000	200	100	2000 × 2000	500	20

In the case of hybrid WMN the number of nodes, coverage area and radio ranges are same as that of client WMN

**Table 3** Simulation parameters

Physical layer protocol	PHY 802.11g
Traffic model of sources	Constant bit rate (CBR)
Mobility model	Random waypoint
Propagation model	Two-ray ground reflection
Antenna	Omni directional
Routing protocol	Dynamic routing
Radio propagation range	200, 250, 350 and 500 m
Simulation trials	20

**Table 4** Results for client WMNs (20 trials for each set were conducted)

No. of nodes	Available computing time (s)	ILC computed by various approaches			
		ACO	BB-BC	Firefly algorithm	Best (shortest) path
10	0.01	1.6795	1.2788	1.2788	0.9012
	0.02	1.6795	1.2788	1.1199	
	0.05	1.6795	1.2788	1.1199	
	0.07	1.1199	0.9012	0.9012	
	0.10	1.1199	0.9012	0.9012	
20	0.01	1.8354	1.8354	1.5938	0.6890
	0.02	1.8354	1.8354	1.4426	
	0.05	1.8354	1.8354	1.0385	
	0.07	1.6119	1.0385	0.6890	
	0.10	1.4426	0.6890	0.6890	
30	0.01	4.0730	3.1848	2.8694	1.4966
	0.02	4.0730	2.7409	2.3926	
	0.05	2.9875	2.1035	1.7653	
	0.07	2.0996	1.6771	1.6771	
	0.10	2.0996	1.4966	1.4966	
50	0.05	4.0108	2.8545	2.2047	1.0273
	0.07	4.0108	1.9273	1.2182	
	0.10	1.9086	1.2182	1.0273	
	0.20	1.7014	1.0273	1.0273	
	0.50	1.2182	1.0273	1.0273	
100	0.05	7.1200	5.0715	4.8389	2.8048
	0.07	5.0715	4.3075	3.8739	
	0.10	4.1214	4.0414	2.8048	
	0.20	3.8739	2.9002	2.8048	
	0.50	3.0464	2.8048	2.8048	

**Table 5** Results of DSR and AODV approaches for client WMNs

	10 Nodes	20 Nodes	30 Nodes	50 Nodes	100 Nodes
<i>ILC computed (time)</i>					
DSR	0.9012 (0.182503 s)	0.6890 (0.215966 s)	1.4966 (0.304508 s)	1.0273 (0.565027 s)	2.8048 (0.699492 s)
AODV	4.5957 (0.316839 s)	7.2174 (0.345390 s)	5.0057 (0.345619 s)	6.8888 (0.651287 s)	11.4412 (0.813981 s)
Best path	0.9012	0.6890	1.4966	1.0273	2.8048

1 acts as source and transmits data packets to the last node which is the terminal node (e.g. 10th node is the terminal node in a 10 node WMN). The data transmission is made possible through multiple hops via various adjacent nodes. In this type of wireless communication multiple routes/paths are available. Decision as regard to which path or route is to be used

**Table 6** Results for hybrid WMNs (20 trials for each set were conducted)

No. of nodes	Time (s)	ILC computed by various approaches			
		ACO	BB-BC	firefly algorithm	Best (shortest) path
10	0.01	2.8900	2.3883	2.0286	1.5132
	0.02	2.8900	2.0286	2.0286	
	0.05	2.5069	1.5132	1.5132	
	0.07	2.3883	1.5132	1.5132	
	0.10	2.0286	1.5132	1.5132	
20	0.01	3.6309	2.1293	2.9493	1.2820
	0.02	3.6309	2.1293	2.9040	
	0.05	3.6309	2.1293	2.8728	
	0.07	3.6309	2.0615	1.2820	
	0.10	2.1293	1.2820	1.2820	
30	0.01	8.0225	3.6633	2.7894	1.7750
	0.02	8.0225	2.2988	2.2988	
	0.05	3.0501	2.2712	2.2816	
	0.07	3.0501	2.2712	1.7750	
	0.10	3.0501	1.7750	1.7750	
50	0.05	4.1577	1.6383	1.4140	0.8065
	0.07	1.5003	1.0163	1.0388	
	0.1	1.5003	1.0163	1.0163	
	0.2	1.0388	0.9728	0.8065	
	0.5	1.0163	0.8065	0.8065	
100	0.05	3.2777	1.9026	1.1591	0.3737
	0.07	3.2777	1.7428	1.0935	
	0.10	1.9586	1.2723	0.9656	
	0.20	1.5467	1.0081	0.3737	
	0.50	1.1591	0.3737	0.3737	

**Table 7** Results of DSR and AODV approaches for hybrid WMNs

	10 Nodes	20 Nodes	30 Nodes	50 Nodes	100 Nodes
<i>ILC computed (time)</i>					
DSR	1.5132 (0.192452 s)	1.2820 (0.374894 s)	1.7750 (0.429841 s)	0.8065 (0.578741 s)	0.3737 (0.826354 s)
AODV	3.8845 (0.273566 s)	6.7321 (0.416487 s)	9.2437 (0.462628 s)	5.1672 (0.665804 s)	8.4266 (1.032847 s)
Best path	1.5132	1.2820	1.7750	0.8065	0.3737

for any type of traffic, depends upon the current value of the ILC measure (distance) of the link connecting two adjacent nodes.

The proposed algorithms were implemented in MATLAB along with AODV and DSR for different architectures of WMNs with varying number of nodes, iterations as well as with different radio ranges and areas. The architectural and simulation details are provided

**Table 8** Results of static (Infrastructure) WMNs (20 trials for each set is conducted)

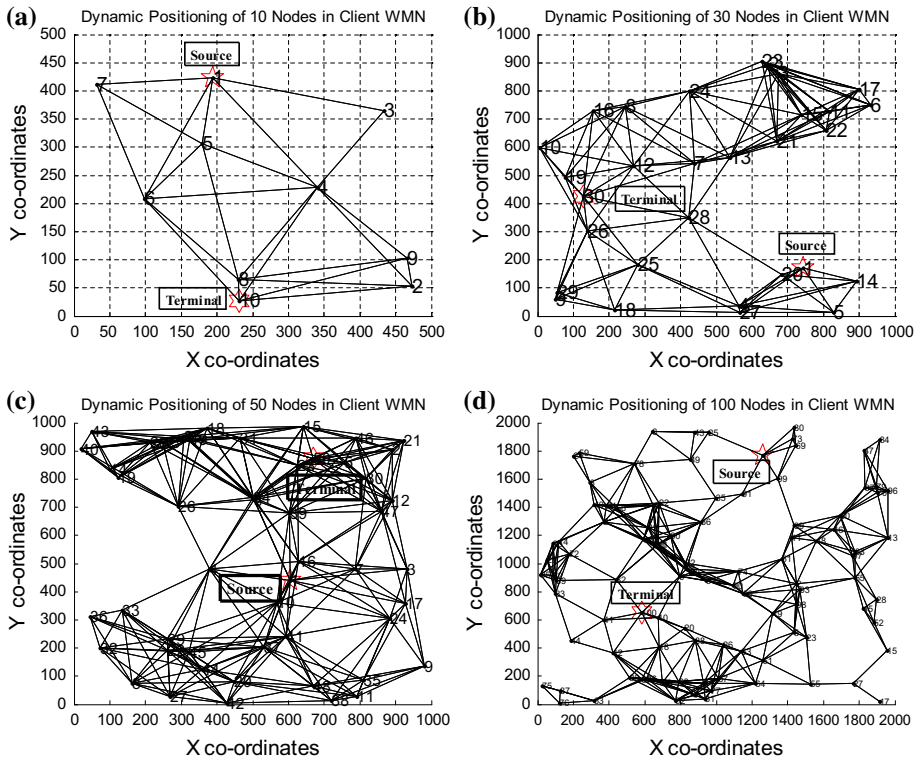
No. of nodes	Time (s)	ILC computed by various approaches			
		ACO	BB-BC	Firefly algorithm	Best (shortest) path
9	0.01	0.8540	0.8540	0.8540	0.6096
	0.02	0.8540	0.8540	0.6096	
	0.05	0.8540	0.6096	0.6096	
	0.07	0.8540	0.6096	0.6096	
	0.10	0.6096	0.6096	0.6096	
16	0.01	1.4276	2.4161	1.4276	1.1173
	0.02	1.4276	2.2876	1.1173	
	0.05	1.4276	2.0416	1.1173	
	0.07	1.1173	1.8928	1.1173	
	0.10	1.1173	1.1173	1.1173	
25	0.01	3.5665	2.3078	1.9605	1.6059
	0.02	3.5665	2.2657	1.6059	
	0.05	3.5665	2.2657	1.6059	
	0.07	3.5665	1.6059	1.6059	
	0.10	2.3078	1.6059	1.6059	
64	0.05	4.5413	3.2500	2.7032	2.1229
	0.07	3.8092	2.7148	2.4079	
	0.10	3.2500	2.6696	2.1229	
	0.20	2.9804	2.5114	2.1229	
	0.50	2.7032	2.1229	2.1229	
100	0.05	4.6037	4.8838	3.5103	2.1848
	0.07	3.9938	3.5103	2.5100	
	0.10	3.6749	3.5103	2.4688	
	0.20	3.2829	2.9948	2.1848	
	0.50	3.1368	2.1848	2.1848	

**Table 9** Results of DSR and AODV approaches for static (infrastructure) WMNs

	9 Nodes	16 Nodes	25 Nodes	64 Nodes	100 Nodes
<i>ILC computed (time)</i>					
DSR	0.6096 (0.165068 s)	1.1173 (0.233253 s)	1.6059 (0.320307 s)	2.1229 (0.632285 s)	2.1848 (0.922432 s)
AODV	3.4533 (0.142666 s)	4.0194 (0.208404 s)	4.8066 (0.308989 s)	10.4955 (0.724616 s)	12.5149 (0.944927 s)
Best path	0.6096	1.1173	1.6059	2.1229	2.1848

in Tables 2 and 3 respectively. The near shortest/shortest path is computed by these five approaches for the same network architecture. The results for different WMN architectures are placed in Tables 4, 5, 6, 7, 8 and 9 for the proposed as well as conventional routing approaches. Each table presents the integrated link cost and processing time for a specific source-terminal node pair for varying number of nodes.





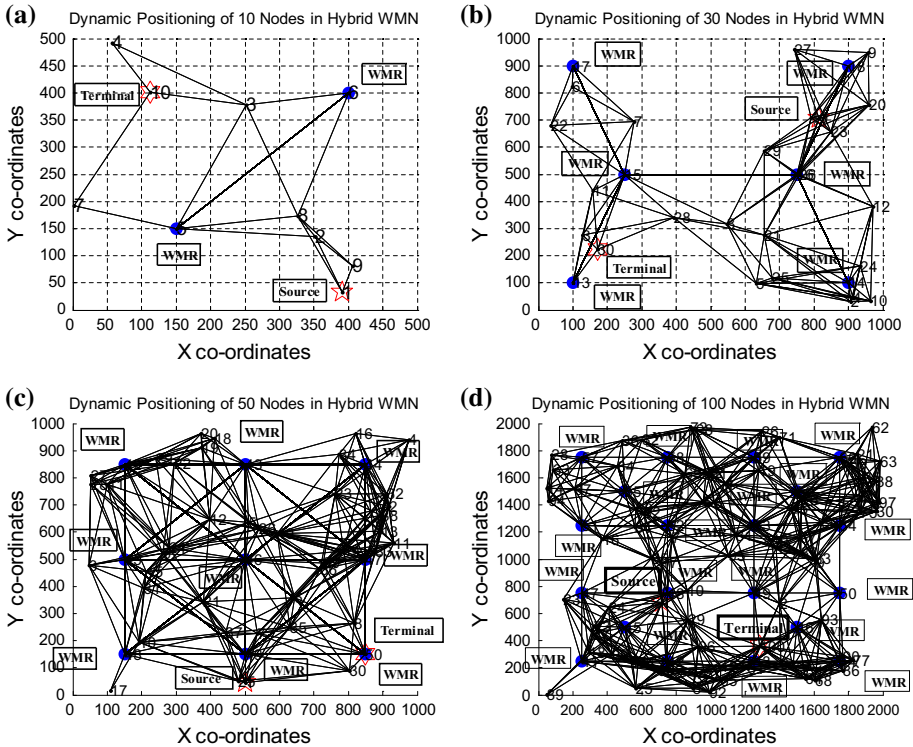
**Fig. 8** Network architecture (a–d) for 10, 30, 50 and 100 node client WMN

In an infrastructure WMN the nodes are placed uniformly in the specified area constructing a uniform mesh. In its routing tables only forward paths are considered to avoid loops. In client and hybrid WMNs, all the nodes are placed randomly in the specified area. This makes the adjacency matrix and hence, routing tables highly dynamic. Hybrid WMNs use some fixed Mesh Routers and there are dedicated links among them. Radio range of these Mesh Routers amongst them is assumed to be double the range of client nodes.

### 5 Results and Discussion

Client, hybrid and infrastructure type WMNs were simulated and the observations were recorded for various network environments. For each of network configurations 20 trials were conducted. The resulting observations are placed as Tables 4, 5, 6, 7, 8 and 9. Table 4 presents the results of the proposed routing approaches for 10, 20, 30, 50 and 100 node client WMNs and Table 5 depicts the results of DSR and AODV algorithms. The results of 10, 20, 30, 50 and 100 node hybrid WMNs are presented in Table 6 for the proposed routing approaches while the resulting observations for DSR and AODV are placed as Table 7. Tables 8 and 9 summarize the results of 9, 16, 25, 64 and 100 node Infrastructure (static) WMNs for the proposed and conventional routing approaches respectively.

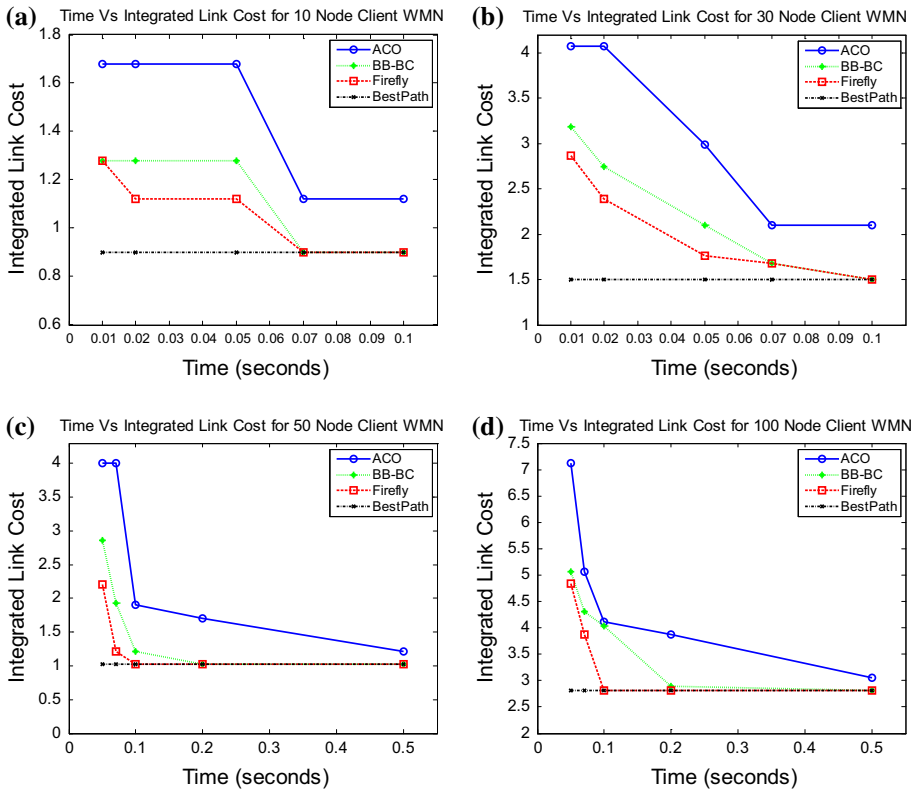
Figure 8 presents specific network architectures of 10, 30, 50 and 100 node client WMNs. It portrays the random behavior of nodes. Figure 9 showcases network



**Fig. 9** Network architecture (a–d) for 10, 30, 50 and 100 node hybrid WMN

architectures of 10, 30, 50 and 100 node hybrid WMNs. WMRs are positioned on specific locations and having dedicated links amongst them. Figures 10 and 11 depict Time versus ILC plots for 10, 30, 50 and 100 node Client and hybrid WMNs respectively. Figure 12 draws the Time versus ILC plot for 9, 25, 64 and 100 node Infrastructure WMNs. For smaller network architectures the proposed algorithms updates the operational phase after an interval of 0.01, 0.02, 0.05, 0.07 and 0.1 s of time constraint while the larger networks are updated after every 0.05, 0.07, 0.1, 0.2 and 0.5 s of time constraint.

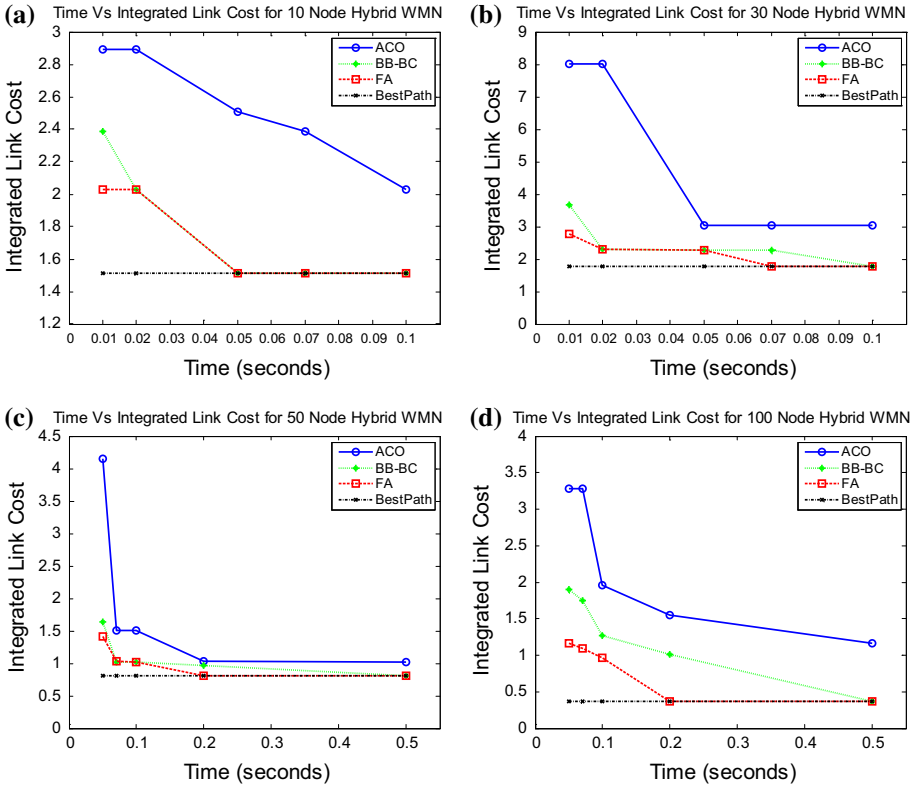
It was observed from all these results that proposed soft computing approaches namely ACO, BB-BC and FA outperform AODV in terms of ILC and processing time. The observations further ascertained that DSR outperforms ACO whereas BB-BC and FA performs better than DSR. In larger networks ACO is unable to converge towards optimal solutions in the given time frame whereas BB-BC and FA converge simply in the stipulated time. FA on the other hand converges much easily and with lesser time followed by BB-BC. As the number of nodes increase processing cost (time) increases accordingly. It was also observed that these proposed meta-heuristic approaches improve their performance with increasing processing time. However a WMN dynamics may allow only small amount of time within which the near shortest path must be computed and routing tables updated. Since, after this allowable time, nodes must have moved to new positions; new shortest or near shortest paths need to be found again.



**Fig. 10** Time versus ILC plot (a–d) for 10, 30, 50 and 100 node client WMN

## 6 Conclusion

This paper proposed three nature inspired routing algorithms for WMNs. The performance of these algorithms was compared with the two existing algorithms namely AODV and DSR. In order to compare the performance we defined a new performance measure namely integrated link cost (ILC). The ILC is based upon the four most significant WMN features namely throughput, delay, jitter and residual energy of the nodes. We applied fuzzy logic to integrate the four network parameters to a single performance measure. Based on these parameters the enumerated ILC was considered to be the distance between two adjacent nodes. Based upon this effective performance metric, the performance of five routing algorithms i.e. Firefly, BB-BC, ACO, DSR and AODV was enumerated between a specific source-terminal node pair. The proposed ILC measure does not affect the scalability and is quite suitable to the stochastic behavior of WMNs. All the five routing approaches were applied to 10, 20, 30, 50 and 100 node client and hybrid WMNs while 9, 16, 25, 64 and 100 node network architecture was considered for infrastructure (static) WMNs with various network topologies. The proposed soft computing based routing approaches were simulated for 0.01, 0.02, 0.05, 0.1, 0.2 and 0.5 s allowable time interval (constraint) for the same network topology. Effect of allowable computing time over ILC was recorded for the proposed soft computing based routing algorithms.



**Fig. 11** Time versus ILC plot (a–d) for 10, 30, 50 and 100 node hybrid WMN

After the extensive simulations runs, we observed that Firefly optimization based routing strategy gives the best routing performance followed by BB-BC, DSR, ACO and lastly AODV. We further observed that Firefly, BB-BC and DSR based routing approaches yield almost similar results in terms of ILC however BB-BC and DSR are taking more time to select the optimized path compared to FA based approach. The ACO based routing algorithm required more computing time and is slower in converging to the optimal solution. Its performance is better than AODV only. For smaller networks, Firefly and BB-BC based approaches converge to reasonably good solutions in the stipulated time frame. ACO based approach was not able to converge to any reasonably good solution within given time frame. Hence, we find that this approach does not meet rigid WMN routing requirements adequately. Since, AODV and DSR performance does not meet WMN requirements at all these do not provide required solution within given time frame we are of opinion that these should be avoided for WMNs (Tables 4, 5, 6, 7, 8, 9). For large WMNs Firefly based routing approach outperforms other routing strategies in terms of ILC and computation time, making FA based approach most suitable followed by BB-BC based approach of the 5 approaches evaluated in this paper.

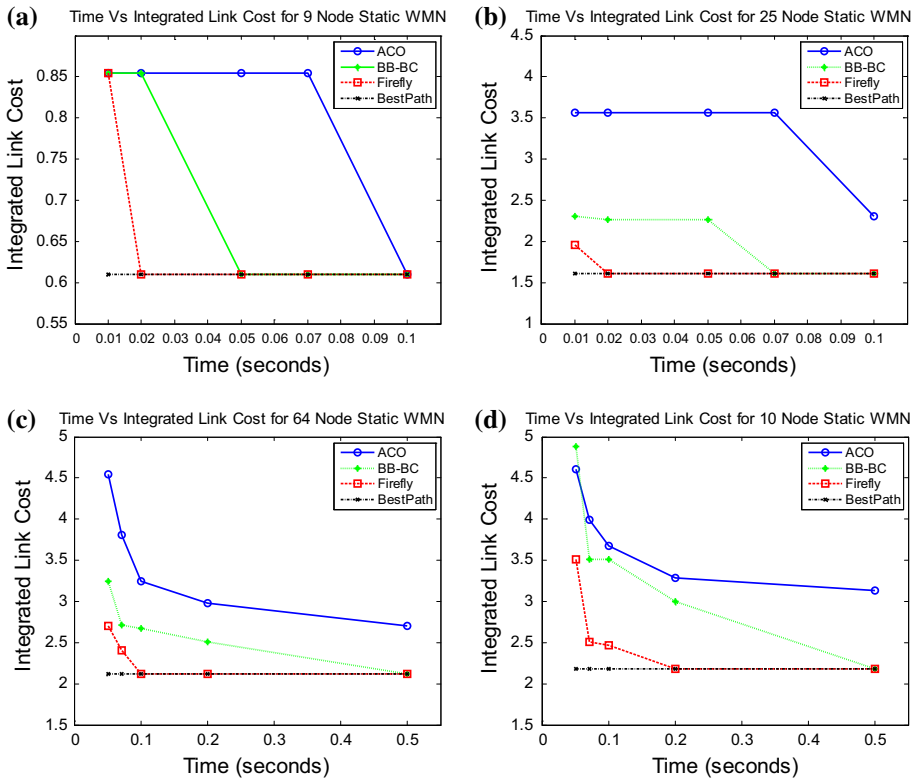


Fig. 12 Time versus ILC plot (a–d) for 9, 25, 64 and 100 node Static WMN

**Acknowledgments** The authors are gratefully acknowledging the comments suggested by the reviewers to improve the technical contents of the paper.

## References

1. Akyildiz, I. F., Wang, X., & Wang, W. (2005). Wireless mesh networks: A survey. *Computer Networks Journal (Elsevier)*, 47(4), 445–487.
2. Bruno, R., Conti, M., & Gregori, E. (2005). Mesh networks: Commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3), 123–131.
3. Parissidis, G., Karaliopoulos, M., Baumann, R., Spyropoulos, T., & Plattner, B. (2009). Routing metrics for wireless mesh networks. In S. Misra, S. C. Misra, & I. Woungang (Eds.), *Guide to wireless mesh networks* (pp. 199–230). London: Springer.
4. Zhang, Y., Luo, J., & Hu, H. (2006). *Wireless mesh networking: Architectures, protocols and standards*. Auerbach Publications.
5. Decouto, D., Aguayo, D., Chambers, B., & Morris, R. (2002). Performance of multi-hop wireless networks: Shortest path is not enough. In *Proceedings of first workshop on hot topics in networks (HotNets-1)*.
6. Waharte, S., Boutaba, R., Iraqi, Y., & Ishibashi, B. (2006). Routing protocols in wireless mesh networks: Challenges and design considerations. *Multimedia Tools and Applications*, 29(3), 285–303.
7. Yang, X. S. (2008). *Nature-inspired metaheuristic algorithms*. Luniver Press.

8. Yang, S., Cheng, H., & Wang, F. (2010). Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 40(1), 52–63.
9. Narendran, R., & Mala, C. (2012). Optimization of QoS parameters for channel allocation in cellular networks using soft computing techniques. *Advances in Intelligent and Soft Computing*, 130, 621–631.
10. Benyamina, D., Hafid, A., Hallam, N., Gendreau, M., & Maureira, J. C. (2012). A hybrid nature-inspired optimizer for wireless mesh networks design. *Computer Communications*, 35(10), 1231–1246.
11. Di Caro, G., Ducatelle, F., & Gambardella, L. M. (2005). Swarm intelligence for routing in mobile adhoc networks. In *Proceedings of IEEE swarm intelligence symposium (SIS-2005)* (pp. 76–83), Pasadena, CA .
12. Dorigo, M., & Blum, C. (2005). Ant Colony Optimization theory: A survey. *Theoretical Computer Science*, 344, 243–278.
13. Dorigo, M., & Stutzle, T. (2004). *Ant Colony Optimization*. Cambridge, MA: MIT Press.
14. Farooq, M., & Di Caro, G. (2008). Routing protocols for next-generation intelligent networks inspired by collective behaviors of insect societies. In C. Blum & D. Merkle (Eds.), *Swarm intelligence: Introduction and applications, natural computing series* (pp. 1–60). Berlin: Springer.
15. Perkins, C., Belding, E., & Das, S. (2003). *Ad hoc on-demand distance vector (AODV) routing*. IETF RFC 3561.
16. Johnson, D. B., Maltz, D. A., & Hu, Y. C. (2004). *The dynamic source routing protocol for mobile ad hoc networks (DSR)*. IETF.
17. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338–353.
18. Gomes, R. L., Moreira, W, Jr, Cerqueira, E., & Abelem, A. J. (2011). Using fuzzy link cost and dynamic choice of link quality metrics to achieve QoS and QoE in wireless mesh networks. *Journal of Network and Computer Applications*, 34(2), 506–516.
19. Chen, Y. L., & Lai, H. P. (2014). A fuzzy logical controller for traffic load parameter with priority-based rate in wireless multimedia sensor networks. *Applied Soft Computing*, 14, 594–602.
20. Chen, Y. L., & Lin, J. S. (2012). Energy efficiency analysis of a chain-based scheme via intra-grid for wireless sensor networks. *Computer Communications*, 35(4), 507–516.
21. Chen, Y. L., Wang, N. C., Shih, Y. N., & Lin, J. S. (2014). Improving low-energy adaptive clustering hierarchy architectures with sleep mode for wireless sensor networks. *Wireless Personal Communications*, 75(1), 349–368.
22. Yen, J., & Langari, R. (1999). *Fuzzy logic intelligence, control and information*. New Jersey: Prentice Hall.
23. Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(1), 28–44.
24. Anderson, D. H., & Hall, L. O. (1999). MR.FIS: Mamdani rule style fuzzy inference system. In *IEEE international conference on systems, man, and cybernetics* (vol. 5, pp. 238–243).
25. Erol, O. K., & Eksin, I. (2006). A new optimization method: Big Bang-Big Crunch. *Advances in Engineering Software*, 37, 106–111.
26. Kripka, M., & Kripka, R. M. L. (2008). Big Crunch optimization method. In *International conference on engineering optimization*, Brazil.
27. Kumar, S., Kaur, P., & Singh, A. (2011). Fuzzy rulebase generation from numerical data using Big Bang-Big Crunch optimization. *Journal of The Institution of Engineers IE (I)*, 91, 18–25.
28. Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-Inspired Computation*, 2(2), 78–84.
29. Yang, X. S. (2010). Firefly algorithm, levy flights and global optimization. In M. Bramer, R. Ellis, & Petridis (Eds.), *Research and development in intelligent systems XXVI* (pp. 209–218). London: Springer.
30. Kumar, S., Kaur, P., & Singh, A. (2012). Fuzzy model identification: A firefly optimization approach. *International Journal of Computer Applications*, 58(6), 1–8.



**Dr. Sharad Sharma** received his B.Tech in Electronics Engineering from Nagpur University, Nagpur, India in 1998 and M.Tech in Electronics and Communication Engineering from Thapar Institute of Engineering and Technology, Patiala, India in 2004. He received his Ph.D. degree in Electronics and Communication Engineering from National Institute of Technology, Kurukshetra, India in 2015. He has a teaching experience of 16 years. He has conducted many workshops on Soft Computing and its applications in engineering, Wireless Networks, Simulators etc. He has a keen interest in teaching and implementing the latest techniques related to wireless and mobile communications. He has opened up a student chapter of IEEE as Branch Counselor. His research interests are routing protocol design, performance evaluation and optimization for wireless mesh networks using nature inspired computing. He is member of various professional bodies such as IEEE, IACSIT and Internet Society.



**Prof. (Dr.) Shakti Kumar** received his M.S. (Electronics and Control) from Birla Institute of Technology and Science (BITS), Pilani and Ph.D. (Electronics and Computer Engg.) from National Institute of Technology (Formerly REC), Kurukshetra. He has served as a faculty member in the Department of Electronics and Computer Engineering, NIT Kurukshetra, Department of Computer Engineering, Atlim University, Ankara, Turkey and Alghurair University, Dubai, UAE. He was Co-ordinator, Centre for Excellence in Reliability Engineering at REC (Now National Institute of Technology), Kurukshetra. He has published more than 165 research articles in various International/National journals and conferences, seven book chapters and has published one patent. He has chaired 13 International/National Conferences. He is member of various professional bodies such as IEEE and ISTE. Dr. Kumar was jointly awarded "Sir Thomas Ward Memorial" Subject Award by Institution of Engineers (India) in December 2006.



**Dr. Brahmjit Singh** received his B.E. Degree in Electronics Engineering from Malaviya National Institute of Technology (then MREC), Jaipur in 1988, M.E. with specialization in Microwave and Radar from Indian Institute of Technology, Roorkee (then UoR) in 1995 and Ph.D. from Guru Govind Singh Indraprastha University, Delhi in 2005. He is with Electronics and Communication Engineering Department, National Institute of Technology working as Professor since 2006. He has published 75 research papers in International/National journals and conferences. His current research interests include mobility management in mobile cellular networks, security in wireless networks, wireless sensor networks and cognitive radio. He has been awarded "Sir Thomas Ward Memorial" Subject Award for the paper titled, "Optimizing Handover Performance in Microcellular Systems" on behalf of Institution of Engineers (India) in December, 2006. He is the member of IEEE, IETE and ISTE.