Middle-East Journal of Scientific Research 24 (Techniques and Algorithms in Emerging Technologies): 249-260, 2016
ISSN 1990-9233;

DOI: 10.5829/idosi.mejsr.2016.24.TAET23534

# Improving Performance of Fuzzy Based Routing Through Optimization for Manet

*[1]R. Menaka, [2]V. Ranganathan and [3]B. Sowmya*

[1]ECE Department, Dhanalakshmi College of Engineering, Chennai,
Research Scholar, Anna University, Chennai, Tamil Nadu, India
[2]Professor and Head/ Electronics and Communication Engineering Department,
Vignan University, Guntur andhra Pradesh, India
[3]Professor, Centre for Research, Dhanalakshmi College of Engineering, Chennai, Tamil Nadu, India

**Abstract:** In ad hoc networks, nodes act as routers and transmit data packets originating from other nodes to destinations beyond the transmission range of the source node. Routing protocol is designed for Mobile Ad hoc Networks (MANET) which assume that all nodes in the network are trustworthy. Security of routing protocol is challenging in realistic setting as networks are attacked by untrustworthy nodes which disrupt the communication. In this paper examines the applicability of Binary Particle Swarm Optimization (BPSO) algorithms in the simultaneous design of membership functions and rule sets for fuzzy logic controllers. Previous work using BPSO has focused on the development of rule sets or high performance membership functions; however, the interdependence between these two components suggests a simultaneous design procedure would be a more appropriate methodology. When BPSO develops both designing membership and rulesets, it is performed serially to enhance results. However, this can only means that membership functions are optimized for initial rule sets and not the rule set designed subsequently. The results show that the MOPSOFR-DSR in untrustworthy environment method increased throughput by 0.46%, 1.76% & 4.12% for FR-DSR method, 1.50%, 1.61% & 1.41% for FR-DSR in untrustworthy environment method and 1.44%, 3.13% & 5.59% for MOPSOFR-DSR method when compared with 15 kmph, 45 kmph & 75 kmph node mobility.

**Key words:** Mobile Ad hoc Networks (MANET) • Routing • Fuzzy Logic • Fuzzy Set • Fuzzy Membership Function • Binary Particle Swarm Optimization (BPSO)

## INTRODUCTION

MANET, a form of wireless networking, has a rudimentary infrastructure pattern multi-hop is created from a set of mobile nodes which unianimously work together to maintainstable connectivity of networks. But, the absence of dedicated fixed routers in networks calls for dual role based structures like MANET, which can add up as a node and a router effectively. MANETs can be rapidly deployed in multiple scenarios due to their lack of structure in scenarios. Examples of such scenarios include disaster recoveries, official business conferences, collaborations, arid region communications etc., where it is impractical and not cost friendly to establish proper infrastructure networks. Routing protocols must maintain communication between nodes in the presence of such abrupt changes in the network due to node failures or mobility. Orthodox infrastructure networks have routing protocols which function on special nodes optimised purposely to engage with resources like procssing power, energy and memory , etc. Routing protocols in MANETs unlike orthodox methods function on normal resource-constrained nodes and form a highly dynamic an unpredictable topology. Mobile nodes are able to directly communicate with one another through MANET when they are in close range to one another and can transmit data to other non-neighbouring nodes along a series of intermediate nodes that acts as routers [1].

Binary sets, a foundational base for modern computer technoloy is a beautiful concept but like the logic of Boolean theories, are unable to model true cognitive and thinking process. Due to its rigid boundaries, the two valued logic are not effective in mapping actual world problems. A solution to real world

**Corresponding Author:** R. Menaka, ECE Department, Dhanalakshmi College of Engineering, Chennai, Research Scholar, Anna University, Chennai, Tamil Nadu, India.

problems was introduced by Zadeh called the concept of 'mathematics of fuzzy or cloudy quantities' followed by his seminal paper 'Fuzzy sets". A subset of Boolean logic, fuzzy mathemathics uses fuzzy rules as one of the important applications of the fuzzy theory. Thus, it is described as a mathematical system that uses analog input value between 0 and 1 in contrast to digital logic [2].

These sets are known to not contain a defined boundary and it holds a degree of membership control over elements. Fuzzy set is a pair (v, m), where v is a set & m: v→ [0, 1]. Fuzzy set theory assesses the membership function of elements in a set which is described by the help of membership function in the interval [0, 1] [3]. Membership Function is the curve or square graph which defines the mapping of each input point to membership value between 0 and 1. Thus, if-then rules can form statements which are capable of creating conditions which upheld fuzzy logic requirements. Singular fuzzy rule comprise of: If x is A then y is B, where A& B are values defined by fuzzy sets on the range x & y respectively. The if part of the rule states x is A and is called as antecedent and then part of the rule is y is B and is called as consequent.

The process of optimization is searching a vector in a function that produces an optimal solution. All of feasible values are available solutions and the extreme value is optimal solution. In general, optimization algorithms are applied to solve these optimization problems. A simple classification way for optimization algorithms is considering the nature of the algorithms and optimization algorithms can be divided into two main categories: deterministic algorithms and stochastic algorithms. Deterministic algorithms using gradients such as hill climbing have a rigorous move and will generate the same set of solutions if the iterations commence with the same initial starting point [4].

On the other hand, stochastic algorithms without using gradients often generate different solutions even with the same initial value. However, generally speaking, the final values, though slightly different, will converge to the same optimal solutions within a given accuracy. Generally, stochastic algorithms have two types: heuristic and meta-heuristic. Recently, nature-inspired meta-heuristic algorithms perform powerfully and efficiently in solving modern nonlinear numerical global optimization problems. To some extent, all meta-heuristic algorithms strive for making balance between randomization (global search) and Local Search (LS).

In general, fuzzy optimization problems are concerned with the maximization or minimization of a single or multiple objectives while satisfying the problem constraints, which represent the model limited resources. Fuzzy optimization's main aim is to find the most satisfying solution (decision alternative) within a fuzzy environment.

To understand and solve a complex problem under a fuzzy environment effectively, two tasks should be accomplished, i.e. fuzzy modelling and fuzzy optimization. Fuzzy modelling aims to build an appropriate model based upon the understanding of the problem and analysis of the fuzzy information, whereas the fuzzy optimization aims at solving the fuzzy model `optimally' by means of optimization techniques and tools on the basis of formulation of the fuzzy information in terms of their membership functions and/or possibility distribution functions, etc. Generally speaking, these tasks represent two different processes; however, there are no precise boundaries between them. The whole process for applying fuzzy optimization to solve a complex problem can be decomposed into seven stages as follows [5]:

- Understanding the problem.
- Fuzziness analysis.
- Development of fuzzy model.
- Description and formulation of the fuzzy information.
- Transformation of the fuzzy optimization model into an equivalent or an approximate crisp optimization model.
- Solving the crisp optimization model.
- Validity examination.

Among the above sub-stages, indicate that the basic procedure of fuzzy optimization is to transform a fuzzy model into a deterministic/crisp optimization one and the most important task is how to make this transformation. During the transformation, the first thing to do is to understand the problem and then to determine the type of optimal solution, e.g. a deterministic solution or a fuzzy solution, according to the understanding. Then, an appropriate interpretation and some concepts for supporting the understanding and definition of the optimal solution are proposed and finally a transformation approach can be developed based on the interpretation. The selection of a particular approach to a fuzzy optimization problem depends on several factors including the nature of the problems, decision-maker's preference and the ranking of the objective as well as its evaluation.

In this work, the fuzzy membership function is optimized using BPSO. Rest of this paper organized as follows: Section 2 discusses the various related works in literature and Section 3 explains methods used in this section. Results and conclusions of this work are discussed in section 4 and 5 respectively.

**Literature Review:** Wang and Huang [6] proposed a scheme based on reputation, hop count and bandwidth to solve the security related problems in adhoc networks. The proposed scheme employs fuzzy logic for selecting best path and showed Quality of Service (QoS) improvement over existing approaches. Tajeddine *et al*., [7] presented a comprehensive model PATROL-F for reputation based trust. This method incorporated a fuzzy subsystem to protect interacting hosts in distributed system. The proposed model implements the concepts of similarity, popularity, activity and cooperation among hosts. It also gives importance to subjective concepts like transaction, the decision in the uncertainty region and setting the result of interaction. The PATROL-F was simulated and the correctness and reliability proved.

Sethi and Udgata [8] proposed a novel approach called Fuzzy-based Trusted Ant Routing (FTAR) using fuzzy logic and swarm intelligence to select its optimal level of path dependence by understanding the various objectives of optimization. This process retains swarm intelligence algorithm advantages and enables trusted routing protocol after applying fuzzy logic systems. Trust values for nodes in MANET are calculated using dropped packat trust-evaluation scheme and Time-Ratio parameters which aim to differentiate among healthy and malicious nodes. Thus, FTAR will consider not only the shortest paths but also high trust levels between neighbors or intermediate nodes.

Santhi and Nachiappan [9] described an idea of selecting best paths from source to destination node in MANETs using fuzzy cost. Here the multicast routing was performed by selecting the most effective path in terms of minimum fuzzy cost by considering multiple independent QoS metrics such as bandwidth, end to end delay and number of nodes to transfer data from the source to the destination. In this method, the available resources of a path are converted into a single metric fuzzy cost. The fuzzy cost was calculated based on multi criterion objective fuzzy measure. The proposed fuzzy cost method was evaluated and compared with conventional protocol MAODV. Results from the above simulations depict the proposed system to have a better hand than conventional MAODV especially in terms of improving packet delivery ratio and minimizing the end to end delay. The proposed multicasting protocol was simulated using the NS-2, while the fuzzy cost was calculated using Matlab.

Gupta *et al*., [10] proposed a routing algorithm based on Fuzzy Logic, which includes minimum communication overhead and storage requirements. The proposed algorithm considers three input variables namely signal power, mobility and delay. The absolute value of each parameter can take a large range at different points on the network.

Santhi and Nachiappan [11] proposed a Fuzzy cost based Multi constrained Quality of service Routing (FCMQR) protocol to select an optimal path by considering multiple independent QoS metrics such as bandwidth, end-to-end delay and number of intermediate hops. The method has amulti criterion objective basis to analyse fuzzy measure and to calculate every available resources in the path which can be converted into singlularlymetric fuzzy cost. Lifeline paths are predicted by Mobility value. Maximum lifetime path and minimum fuzzy cost are considered as optimal solutions to be used for transmission. Simulation results show that the proposed FCMQR provides an accurate and efficient method of estimating and evaluating the QoS routing stability and cost in dynamic mobile networks.

Pi and Sun [12] introduced a fuzzy controllers based multipath routing algorithm in MANET (FMRM). The key idea of FMRM algorithm was to be constructed the fuzzy controllers with the help to be reduced modifications within adhoc network. Simulation results propose that the above approach is effective and efficient in applications to the MANETs. It was an available approach to multipath routing decision.

Yuste *et al*., [13] presented a fuzzy logic system which should be installed in the mobile nodes to distributedly identify the stable routes. In particular, the system was supported by an interval-based type-2 fuzzy logic. Being a type-2 fuzzy logic system, it was able to cope with inexact estimations. This ability important to be avoided as using additional messages occupy a largescale wireless medium. But, interval-based fuzzy system can provide the required simplicity which is often demanded by mobile devices which energy-constrained. The two outputs within interval-based fuzzy systems are employed because of their novelty. The use of each output depends on the traffic state of the mobile node. By means of extensive simulations, it demonstrated that the goodness of the proposed system.

## MATERIALS AND METHODS

In this section, the fuzzy membership function is optimized using BPSO algorithm are described.

**Fuzzy Membership Function:** Fuzzy systems are obtained from the desire for describing a complex system containing linguistic descriptions in it. While Boolean systems allow an item to have a membership of either one or zero in a set, fuzzy systems allow for degrees of membership over the range [0, 1]. This imitates the linguistic, non-precise approach to describing conditions (i.e., cold, very warm) used in everyday life. Fuzzy controllers allow for a simpler, more human approach to control design and do not demand the mathematical modelling knowledge of more conventional control design methods. The ability is mathematically describe systems become difficult when complexities increase within systems. For this reason, fuzzy controllers provide reasonable, effective alternatives to classical or state-space controllers [14].

Proposed fuzzy logic systems can implements basic human experiences and accordingly decide on preferences based on the set characters in membership functions and fuzzy rules. Fuzzy logic systems are proposed to find Link Stability Coefficients (LSC) within each link especially among its source and destination. It often uses two input variables and a singular outputs. Both the inputed variables can be fuzzified are Δd and Δv among neighbor nodes. Inputs are then implicated, fuzzified, defuzzified and aggregated to acquire a crisp value format of LSC as stipulated outputs. Linguistic variable are often associated with input variables which can be defined as Low (L), Medium (M) and High (H) for Δd and Negative (N), Zero (Z) and Positive (P) for Δv. For the output variable, link stability index, six linguistic variables are used. They are, Very Low (VL), L, M, Average (A), High (H) and ofcourse, the Very High (VH) values. The above membership functions are designed to be triangular. Fuzzy conditional rules of fuzzy stability are shown in table 1. The first rule can be interpreted as, "If (Δd is low) and (Δv is negative) then link stability is medium". Similarly the other rules have been developed [15].

Table 1: Fuzzy conditional rules

| $\Delta v$ / $\Delta d$ | N | Z | P |
|---|---|---|---|
| L | M | VH | A |
| M | L | H | A |
| H | VL | A | H |

LSC values between every neighboring nodes is computed through fuzzy logic systems. Here, it use $LSC_{i,j}$ denote the LSC between node i and node j. Assume one communication route between source and destination is made up of n intermitted nodes.

$$RSC_{s,d} = LSC_{s,1} * LSC_{1,2} * LSC_{2,3} * ......*LSC_{n,d}$$

Here, the $RSC_{s,d}$ can denote Rout Stability Coefficient of a whole route.

The basic blocks of Fuzzy Logic Controller have been explained as follows [16]:

**The Fuzzification Module:** The above module can convert each crisp input into a set of fuzzy values based on the domain in which input variables are existent. Different kinds of membership have neen proposed for the above purpose such as triangular, Gaussians, sigmoid etc. The membership function used in the proposal is triangular.

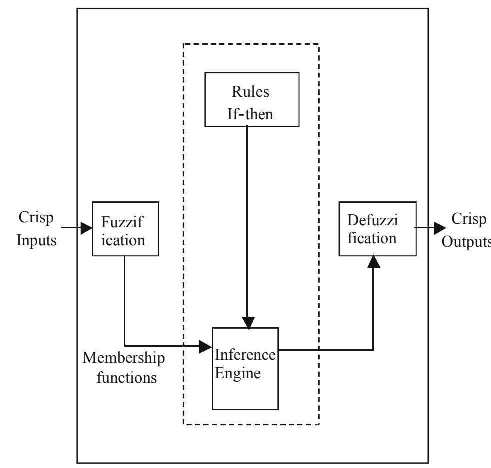


Fig. 1: Fuzzy Logic Controller

**Rule Base:** This module contains rules of the form “IF THEN” in which “IF” is also known as the antecedent and “THEN” is denoted as a consequent. Inference engines function on the above basis.

**Inference Engine:** This is a computer program which attempts to derive answers from specified rule bases. Results from mamdami-type inference engines are calculated by the above program.

**Defuzzification Module:** The above module has the ability to convert fuzzy sets into crisp sets and there are multiple literature available on methods for defuzzifications like the Mean-of-Maxima, Height method and Centre of Gravity (COG), etc.

Fuzzy theory owes a great deal too human language. Daily languages cannot be precisely characterized on either the syntactic or semantic level. When it speak of temperature in terms such as “hot” or “cold” instead of in physical units such as degrees Fahrenheit or Celsius, it can see language becomes a fuzzy variable whose spatial denotation is imprecise. So fuzzy theory becomes easily understood as it can resemble top level languages instead of mathematically coded languages. Fuzzy sets are often named as "hot" or "cold" to describe a universal discourse and create possible membership functiosn [14].

Input among fuzzy sets function after determining the degree of membership function and through this one can see the role membership functions play in decoding the linguistic terminology to the values a computer can use. Of course in most respects these membership functions are subjective in nature. What determines the ranges for these fuzzy-set values or the shape of these membership functions? In most cases, membership functions are designed by experts with a knowledge of the system being analyzed. However, human experts cannot be expected to provide optimal membership functions for a given system. Often, these functions are modified iteratively while trying to obtain optimality. How are these membership functions used in fuzzy controllers? In its simplest form a fuzzy logic controller is simply a set of rules describing a set of actions to be taken for a given set of inputs. It is easiest to think of these rules as if then statements of the form IF {set of inputs} THEN {outputs}.

As an example, consider a fuzzy controller used in the cart controller problem. One rule might be ZF {distance from 0 very far in positive z-direction} and {velocity >> 0) THEN {apply a force <O}. Another rule may be ZF {distance from 0 near in negative z-direction} and (velocity = 0) THEN {apply a force > 0). Since "very far" applies to a range of distances which also may belong to another fuzzy-set variable (i.e., "far") which has rules of its own, the output which results from "defuzzification" of the application of these rules must take into account how much each rule applies before determining how much output must be applied. Usually a centroid method is used to account for the influence of each rule on the output.

Zadeh introduced about the membership functions in the first paper on fuzzy sets (1965). Human experiences are often applied to Fuzzy logic and its preferences are extablished via membership functions and fuzzy rules. A membership function represents a fuzzy set as a mathematical formation. A fuzzy number is a quantity whose value is imprecise, rather than exact as is the case with "ordinary" (single-valued) numbers [17].

For any set X, a membership function on X is any function from X to the real unit interval [0, 1]. Membership functions on X represent fuzzy subset of X. The membership function set is usually denoted by $\mu_A$. For an element x of X, the value $\mu_A(x)$ is called the membership degree of x in the fuzzy set $\mu_A(x)$. mathematically summarizes membership grade of element x to the fuzzy set.

$\mu_A(x) = 0$ means that x is not a member of fuzzy set.
$\mu_A(x) = 1$ means that x is fully member of fuzzy set.
$\mu_A(x)$ between 0 and 1 characterize fuzzy members, which belong to set partially.

The membership functions were designed to satisfy the following two conditions: (1) Each membership function overlaps only with the closest neighbouring membership functions; (2) the membership values of all relavant fuzzy sets are summed as 1 for given input dataset. The membership functions can be specified as follows [18]:

- Residual Energy (RE) is represented by 3 triangular membership functions. The representation is shown in Figure 1. These are specified by three parameters {a, b, c} as follows:

$$triangular(x:a,b,c)=\begin{cases}0 & x<a\\(x-a)/(b-a) & a\le x\le b\\(c-x)/(c-b) & b\le x\le c\\0 & x>c\end{cases}$$

- Mobility is represented by 2 trapezoidal membership functions. A Trapezoidal membership function is specified by four parameters {a, b, c, d} as follows:

$$trapezoid(x:a,b,c,d)=\begin{cases}0 & x<a\\(x-a)/(b-a) & a\le x<b\\1 & b\le x<c\\(d-x)/(d-c) & c\le x<d\\0 & x\ge d\end{cases}$$

- Here, the above traffic is denoted through the 3 triangular membership functions, where a=0, b=50% and c=Tmax. The max bit rate is considered to be 11Mbps (when using 802.11g). All values are normalized with respect to 11 Mbps. The above triangular functions are used to ensure smooth transitions from low, medium and high traffic situations.
- The output is represented by three trapezoidal member functions such as good, acceptable, or bad The smooth transition from bad and acceptable occurs between 5% and 45%, while the smooth transition from acceptable to good occurs between 55% and 95%.

There are mainly six fuzzy rules which are utilized: (1) if RE values are high, then output is considered to be good. (2) A higher mobility OR higher traffic ensures, bad output. (3) If RE is calculated to be medium AND low mobility, then outputs are acceptable. (4) If RE is low, then outputs are bad. (5) If RE is medium AND traffic is medium, then outputs are considered acceptable. (6) If RE is medium AND traffic is low, then output is also considered acceptable. The above rules summarize the CH properties, i.e. the CH is preferred to have high RE, low mobility and low traffic.

**Binary Particle Swarm Optimization (BPSO):** The above optimization technique is an evolutionary method proposed by Kennedy and Eberhart in 1995. Particle Swarm Optimizations affect social behavioural patterns, such as the flocking of birds and schools of fish. A population in PSOs are also known as a swarm of candidate solutions which are basically encoded by particles in a search space. PSOs begin engagements through randomly initialising particle populations. This selected swarm moves along in search space to find the best solution by updating positions of each particle based on the experience of its own and its neighbouring particles [19].

The current position $i$ of particles calculated during movement $t$ is represented by the following vector,

$$x_i = (x_{i1}, X_{i2},......,x_{iD}) \tag{1}$$

In the above equation, D measures the dimensionality of a search space.

The velocity of particle i is represented as,

$$v_i = (v_{i1}, v_{i2},......,v_{iD}) \tag{2}$$

Equation (2) shows the occurrence of limited predefined maximum velocity;

$$v_{\max} \; and \; v_{id}^{i} \in [-v_{\max}, v_{\max}] \tag{3}$$

Particle positons are recorded and amongst this list of previous positions, the personal best is known as pbest and its best position obtained by the population is denoted by gbest. Thus based ponbest and gbest, PSOs continue to search for optimal solution by constantly updating velocities and positions of every particle according to the following formula:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{4}$$

This method was initailly proposed for solving real number search space problems in. But, numerous optimisation problems like feature selections can happen among discrete search spaces. The Particles with pBest and gBest for Optimization as shown in Figure 3 [20, 21]:

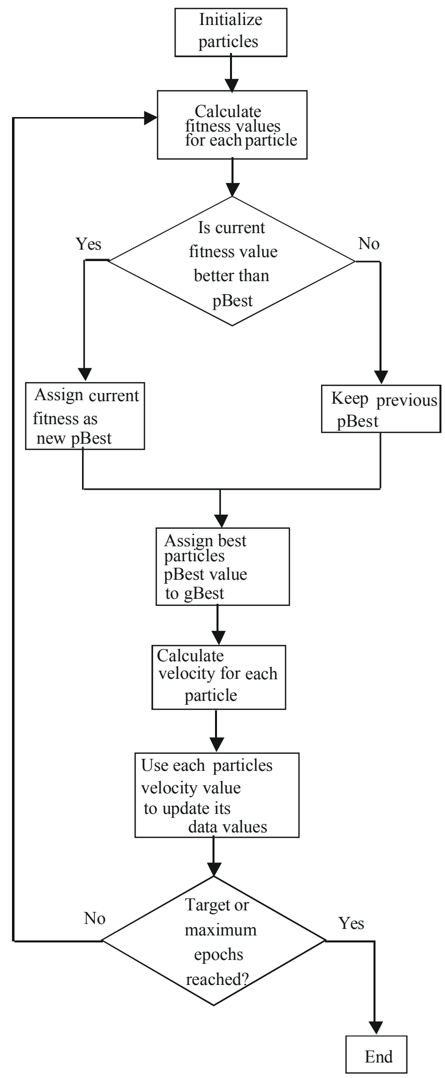


Fig. 2: Particles with pBest and gBest for Optimization

In 1997 the binary version of this algorithm was presented by Kennedy and Eberhart for discrete optimization problems. In this method, each particle has a position in a D-dimensional space and each element of a particle position can take the binary value of 0 or 1 in which 1 means “included” and 0 means “not included”. Major differences between binary PSO exist in continuous versions so that the velocities can be defined in terms of different probabilities. Using this definition a velocity must be restricted within the range [0, 1]. So a map is introduced to map all real valued numbers of velocity to the range [0, 1]. The normalization function used typically is a sigmoid function as [22]:

$$V_k^{t+1} = V_k^t(i) + c_1 r_1 (Lbest^i(t) - X_k^t(i)) + c_2 r_2 (Gbest^i(t) - X_k^t(i)) \tag{5}$$

$$S(V_k^{t+1}(i)) = 1 \big/ (1 + e^{-V_k^{t+1}(i)}) \tag{6}$$

$$X_k^{t+1}(i) = \begin{cases} 1 & if \;\; rand() \le S(V_k^{t+1}(i)) \\ 0 & otherwise \end{cases} \tag{7}$$

From the above equation the values of rand () are drawn using U (0, 1) and functions S (v) which represents a sigmoid limiting transformation. The number of particles in the beginning of the above algorithm and their velocity vectors are randomly generated. Ater which some of the iterations in the algorithm try to obtain optimal or near-optimal solutions , based on its ability to follow predefined fitness function. The velocity vector is updated in each time step using two best positions, Lbest and Gbest and then the position of the particles is updated using velocity vectors.

The BPSO algorithm contains [23]:

1. *Initialize the swarm $X_i$ , the position of particles are randomly initialized within the hypercube. Elements of $X_i$ are randomly selected from binary values 0 and 1.*
2. *Evaluate the performance F of each particle, using its current position $X_i(t)$ .*
3. *Compare the performance of each individual to its best performance so far :*
   $if \;\; F(X_i(t) < F(P_{ibest}))$ :
   $F(P_{ibest}) = F(X_i(t))$
   $P_{ibest} = X_i(t)$
4. *Compare the performance of each particle to the global best particle :*
   $if \;\; F(X_i(t)) < F(P_{gbest})$ :
   $F(P_{gbest}) = F(X_i(t))$
   $P_{gbest} = X_i(t)$
5. *Change the velocity of the particle, $\vec{v}_i^0$ and $\vec{v}_i^1$ according to*
   $V_{ij}^1 = wV_{ij}^1 + d_{ij,1}^1 + d_{ij,2}^1$
   $V_{ij}^0 = wV_{ij}^0 + d_{ij,1}^0 + d_{ij,2}^0$
6. *Calculate the velocity of change of the bits, $\vec{v}_i^c$ as in*
   $$V_{ij}^c = \begin{cases} V_{ij}^1, & if \;\; x_{ij} = 0 \\ V_{ij}^0, & if \;\; x_{ij} = 1 \end{cases}$$
7. *Generate the random variable $r_{ij}$ in the range : (0,1) . Move each particle to a new position using*
   $$x_{ij}(t+1) = \begin{cases} \bar{x}_{ij}(t), & if \;\; r_{ij} < V_{ij}' \\ x_{ij}(t), & if \;\; r_{ij} < V_{ij}' \end{cases}$$
8. *Go to step 2, and repeat until convergence.*

Differences amongst PSO and BPSO are portrayed through their defined search spaces. In PSO, the movement of space meant the change of coordinate position in one or more existing dimensions. However, in the BPSO moving in the spaces means a change in the probability of the fact that the value of position coordinate is zero or one [24].

To avoid the local optimum problem, here (8) and (9) are all reserved for iterative evolution in BPSO and then it adopt a novel formula to determine a binary bit $px_{ij}$., which can be denoted as follows [25]:

$$L(\mathrm{x}_{ij}) = (\mathrm{x}_{ij} - \mathrm{R}_{\min}) / (\mathrm{R}_{\max} - \mathrm{R}_{\min}) \tag{8}$$

$$px_{ij} = \begin{cases} 1 & \text{if rand()<L(x}_{ij}) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

In the above equation , *L(x)* is represented as a linear function with an output value which can be represented between (0, 1); rand () is usually a stochastic number picked from a uniform distribution set ranging from [0.0, 1.0]; and [$R_{max}$, $R_{min}$] as a predefined range used for gaining probability values with L(x) function.

In summary, the flowchart of the BPSO is given in Figure 4, where P is on behalf of the population size, G denotes the maximum iterative times.

BPSO is initialized with initial biclusters which is obtained by using K-Means on both the dimensions of the web access matrix A. This result in fast convergence of the gbest compared to random particle initialization of the BPSO and it also maintains high diversity in the population [26].

Each particle of BPSO has the capability to explore best possible solutions. The flight is adjustable according to its companions flying experience and its own. Personal best (pbest) positions are the best solution found in a particle within its flight phase. Pbest denoted the above particle and also can assist in calulating optimal solution which are attained from the global topper (gbest). BPSO updates iteratively the velocity of each particle towards

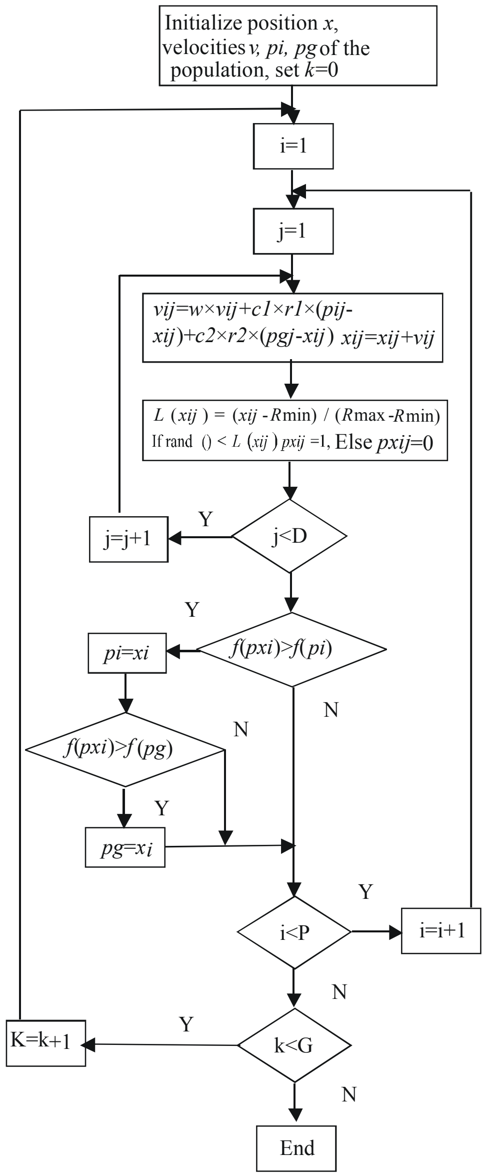


Fig. 3: The flowchart of BPSO

the pbest and gbest. For finding an optimal or near optimal solution to the problem, BPSO contantly update current generations of particles and this is a repetative process until stopping criterions aew met or maximum number of iteration has been achieved.

The fuzzy membership function optimized using BPSO: the initial population, which contains a number of solutions, is generated using a sophisticated priority list. Such an action is undertaken because the random generation of population will probably produce low-quality and infeasible solutions, which in turn may lead the method to premature convergence.

A fitness function is necessary to measure the performance of a solution. Some penalty factors associated with crisp constraints violations and the aggregated fuzzy membership functions for the imprecise constraints are incorporated with the fitness function. The penalty term will be used to discourage the non-potential and erroneous solution and is defined by the margin of constraint violations. The fitness comprises the total production cost, the violated constraints and the overall fuzzy membership degree of the incorporated constraints. Load demand, wind speed, spinning reserve and production cost are fuzzified in the proposed method by assigning membership degree depending on the error margin. In each population, the cluster size is increased to reduce the number of clusters and to increase the convergence speed. Also, applied an intelligent mutation operator with dynamic probability rate to the solutions.

Finally, Eberhart and Shi devised an adaptive fuzzy PSO, where a fuzzy controller was used to control ω over time. This approach is very interesting, since it potentially lets the PSO self-adapt ω to the problem and thus optimizes and eliminates a parameter of the algorithm. This saves time during the experimentation, since fine-tuning of ω is not necessary anymore. At each time-step, the controller takes the "Normalized Current Best Performance Evaluation" (NCBPE) and the current setting of ω as inputs and it outputs a probabilistic change in ω [27].

## RESULTS AND DISCUSSION

In this section, the FR-DSR, FR-DSR in untrustworthy environment, MOPSOFR-DSR and MOPSOFR-DSR in untrustworthy environment methods are evaluated. The throughput, end to end delay and Percentage of Malicious node detected as shown form Table 2 to 4 and Figure 5 to 7.

From the Figure 5, it can be observed that the MOPSOFR-DSR in untrustworthy environment method increased throughput by 0.46%, 1.76% & 4.12% for FR-DSR method, 1.50%, 1.61% & 1.41% for FR-DSR in untrustworthy environment method and 1.44%, 3.13% & 5.59% for MOPSOFR-DSR method when compared with 15 kmph, 45 kmph & 75 kmph node mobility.

From the Figure 6, it can be observed that the MOPSOFR-DSR in untrustworthy environment method decreased End to End Delay by 3.08%, 6.88% & 2.56% for FR-DSR method, 6.03%, 2.30% & 4.23% for FR-DSR in untrustworthy environment method and 6.49%, 10.89% & 2.27% for MOPSOFR-DSR method when compared with 15 kmph, 45 kmph & 75 kmph node mobility.

Table 2: Throughput Achieved

| Node mobility | FR-DSR | FR-DSR in untrustworthy environment | MOPSOFR-DSR | MOPSOFR-DSR in untrustworthy environment |
|---|---|---|---|---|
| No mobility | 0.97 | 0.95 | 0.99 | 0.96 |
| 15 kmph | 0.94 | 0.93 | 0.96 | 0.94 |
| 30 kmph | 0.92 | 0.91 | 0.94 | 0.92 |
| 45 kmph | 0.90 | 0.87 | 0.92 | 0.89 |
| 60 kmph | 0.85 | 0.78 | 0.87 | 0.79 |
| 75 kmph | 0.81 | 0.77 | 0.83 | 0.78 |

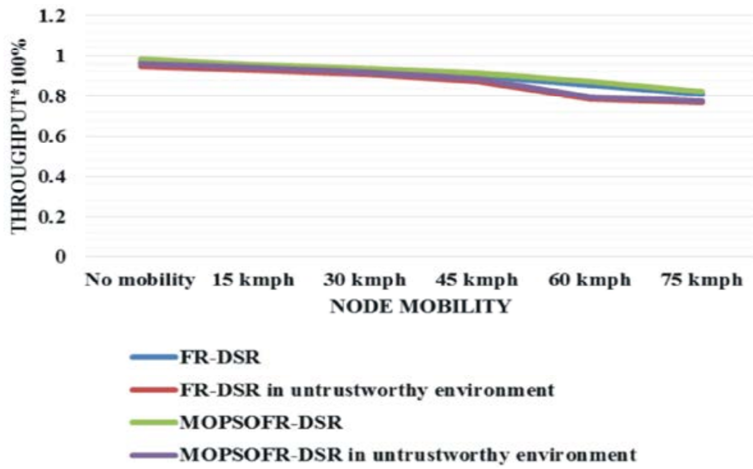


Fig. 4: Throughput Achieved

Table 1: End to End Delay in millisecond

| Node mobility | FR-DSR | FR-DSR in untrustworthy environment | MOPSOFR-DSR | MOPSOFR-DSR in untrustworthy environment |
|---|---|---|---|---|
| No mobility | 0.00173 | 0.00194 | 0.001641 | 0.00189 |
| 15 kmph | 0.00262 | 0.00287 | 0.002532 | 0.002702 |
| 30 kmph | 0.00363 | 0.00376 | 0.003527 | 0.003681 |
| 45 kmph | 0.00384 | 0.00421 | 0.003689 | 0.004114 |
| 60 kmph | 0.00457 | 0.00468 | 0.004307 | 0.00445 |
| 75 kmph | 0.00474 | 0.00482 | 0.004516 | 0.00462 |

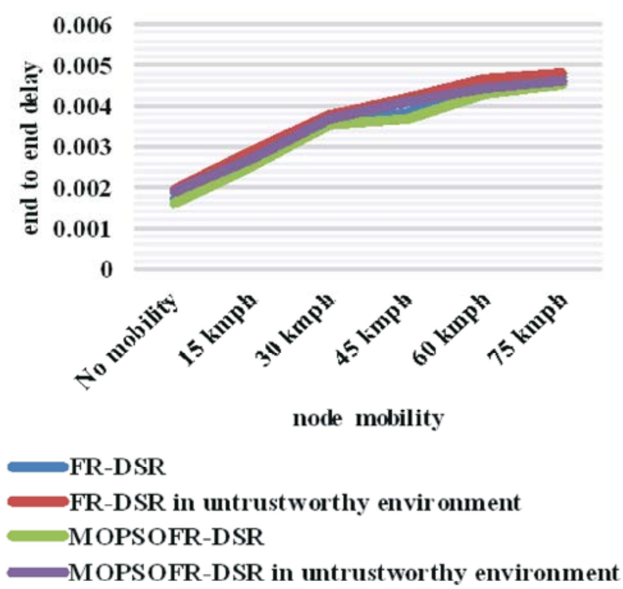


Fig. 5: End to End Delay in millisecond

Table 2: Percentage of Malicious node detected

| | FR-DSR in untrustworthy environment | | MOPSOFR-DSR in untrustworthy environment | |
|---|---|---|---|---|
| Node mobility | Initial state | Steady state | Initial state | Steady state |
| No mobility | 0.4 | 0.9 | 0.5 | 0.9 |
| 15 kmph | 0.4 | 0.8 | 0.4 | 0.9 |
| 30 kmph | 0.3 | 0.8 | 0.4 | 0.8 |
| 45 kmph | 0.3 | 0.7 | 0.3 | 0.7 |
| 60 kmph | 0.2 | 0.6 | 0.3 | 0.7 |
| 75 kmph | 0.1 | 0.4 | 0.2 | 0.5 |

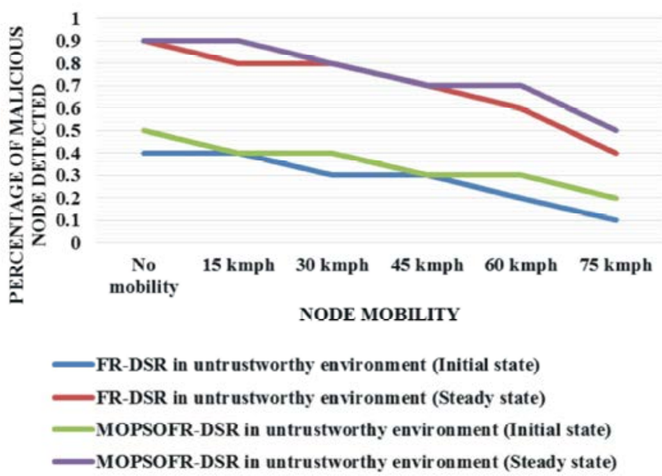


Fig. 6: Percentage of Malicious node detected

From the Figure 7, it can be observed that the MOPSOFR-DSR in untrustworthy environment (Steady state) method increased Percentage of Malicious node detected by 76.92%, 80% & 133.33% for FR-DSR in untrustworthy environment (Initial state) method, 11.76%, 0% & 22.22% for FR-DSR in untrustworthy environment (Steady state) method and 76.92%, 80% & 85.71% for MOPSOFR-DSR in untrustworthy environment (Initial state) method when compared with 15 kmph, 45 kmph & 75 kmph node mobility.

## CONCLUSION

This paper clearly shows the potential for using BPSO algorithms to solve optimization problems. The ability of fuzzy logic controllers to provide control where more conventional methods become too complex has also been shown by researchers. This work has shown these two, fairly new, methods can be used together to form controllers without the previously needed human expertise. This methodology allows the complete design of both major components of fuzzy controllers, the rule sets and membership functions, leading to high performing controllers which are completely computer-designed. It has developed four different controllers for the cart problem, each of which was able to bring the cart to equilibrium over the entire ranges of the input spaces. Also, it has shown that BPSO has the ability to design a robust controller which can work over a wide parameter range. Also, as mentioned earlier, the inclusion of finding the location of the peaks of the triangles in the membership functions will yield even higher performing controllers. Finally, controllers for still more problems should be examined to show the effectiveness of this method. The results show that the MOPSOFR-DSR in untrustworthy environment method increased throughput by 0.46%, 1.76% & 4.12% for FR-DSR method, 1.50%, 1.61% & 1.41% for FR-DSR in untrustworthy environment method and 1.44%, 3.13% & 5.59% for MOPSOFR-DSR method when compared with 15 kmph, 45 kmph & 75 kmph node mobility.

## REFERENCES

1. Ali, M., B.G. Stewart, A. Shahrabi and A. Vallavaraj, 2015. Fuzzy based Load and Energy Aware Multipath Routing for Mobile Ad Hoc Networks. International Journal of Computer Applications, 114(16).

2. Siddiqui, S., P.M. Khan and M.U. Khan, 2014. Fuzzy Logic Based Intruder Detection System in Mobile Adhoc Network. Our Indexing at International Level, 4852, 767.
3. Saju, I. and N. Krishnan, 2014. A Survey On Fuzzy-Based Adaptive Routing Protocol For Manets. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 3(3).
4. Wang, G.G., A.H. Gandomi, X. Zhao and H.C.E. Chu, 2016. Hybridizing harmony search algorithm with cuckoo search for global numerical optimization. Soft Computing, 20(1): 273-285.
5. Chalco-Cano, Y., G.N. Silva and A. Rufián-Lizana, 2015. On the Newton method for solving fuzzy optimization problems. Fuzzy Sets and Systems, 272: 60-69.
6. Wang, J.L. and S.P. Huang, 2007. Fuzzy logic based reputation system for mobile ad hoc networks. In Knowledge-Based Intelligent Information and Engineering Systems (pp: 1315-1322). Springer Berlin Heidelberg.
7. Tajeddine, A., A. Kayssi, A. Chehab and H. Artail, 2011. Fuzzy reputation-based trust model. Applied Soft Computing, 11(1): 345-355.
8. Sethi, S. and S.K. Udgata, 2011. Fuzzy-based trusted ant routing (FTAR) protocol in mobile ad hoc networks. In Multi-disciplinary Trends in Artificial Intelligence, pp: 112-123. Springer Berlin Heidelberg.
9. Santhi, G. and A. Nachiappan, 2011. Fuzzy cost based multicast routing for mobile ad-hoc networks with improved QoS. In Advances in Digital Image Processing and Information Technology, pp: 429-437. Springer Berlin Heidelberg.
10. Gupta, S., P.K. Bharti and V. Choudhary, 2011. Fuzzy Logic Based Routing Algorithm for Mobile Ad Hoc Networks. In High Performance Architecture and Grid Computing, pp: 574-579. Springer Berlin Heidelberg.
11. Santhi, G. and A. Nachiappan, 2012. Fuzzy-cost based multiconstrained QoS routing with mobility prediction in MANETs. Egyptian Informatics Journal, 13(1): 19-25.
12. Pi, S. and B. Sun, 2012. Fuzzy Controllers Based Multipath Routing Algorithm in MANET. Physics Procedia, 24: 1178-1185.
13. Yuste, A.J., A. Triviño and E. Casilari, 2013. Type-2 fuzzy decision support system to optimise MANET integration into infrastructure-based wireless systems. Expert Systems with Applications, 40(7): 2552-2567.
14. Li, H., X. Sun, L. Wu and H.K. Lam, 2015. State and output feedback control of interval type-2 fuzzy systems with mismatched membership functions. Fuzzy Systems, IEEE Transactions on, 23(6): 1943-1957.
15. Das, S.K., S. Tripathi and A.P. Burnwal, 2015. February). Fuzzy based energy efficient multicast routing for ad-hoc network. In Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on, pp: 1-5. IEEE.
16. Singh, P., V. Kukreja, P. Arora and P. Hooda, 2015. A Fuzzy Based Routing Strategy for Mobile Adhoc Networks (MANET). International Journal of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering, 3(7).
17. Dahiya, R. and A. Dureja, 2014. Fuzzy Based Efficient Routing Protocol For Route Recovery In MANET. International Journal Of Engineering And Computer Science, 3(6): 6681-6687.
18. Chettibi, S. and S. Chikhi, 2016. Dynamic fuzzy logic and reinforcement learning for adaptive energy efficient routing in mobile ad-hoc networks.Applied Soft Computing, 38: 321-328.
19. Beheshti, Z., S.M. Shamsuddin and S. Hasan, 2015. Memetic binary particle swarm optimization for discrete optimization problems.Information Sciences, 299: 58-84.
20. Tanweer, M.R., S. Suresh and N. Sundararajan,. 2015. Self regulating particle swarm optimization algorithm. Information Sciences, 294: 182-202.
21. Phoa, F.K.H., R.B. Chen, W. Wang and W.K. Wong, 2015. Optimizing two-level supersaturated designs using swarm intelligence techniques.Technometrics, (just-accepted), 00-00.
22. Robinson, Y.H. and M. Rajaram, 2015. Energy-aware multipath routing scheme based on particle swarm optimization in mobile ad hoc networks.The Scientific World Journal, 2015.
23. Jin, C. and S.W. Jin, 2015. Automatic image annotation using feature selection based on improving quantum particle swarm optimization. Signal Processing, 109: 172-181.
24. Jordehi, A.R., 2015. A review on constraint handling strategies in particle swarm optimisation. Neural Computing and Applications, 26(6): 1265-1275.

25. Chatterjee, A., G.K. Mahanti and A. Mahanti, 2015. Synthesis of thinned concentric ring array antenna in predefined phi-planes using binary firefly and binary particle swarm optimization algorithm. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 28(2): 164-174.
26. Cai, Q., M. Gong, L. Ma, S. Ruan, F. Yuan and L. Jiao, 2015. Greedy discrete particle swarm optimization for large-scale social network clustering. Information Sciences, 316: 503-516.
27. Ma, W., M. Wang and X. Zhu, 2015. Hybrid particle swarm optimization and differential evolution algorithm for bi-level programming problem and its application to pricing and lot-sizing decisions. Journal of Intelligent Manufacturing, 26(3): 471-483.