Towards Green Communication in Wireless Sensor Network: GA Enabled Distributed Zone Approach

Sushil Kumar [a], Vipin Kumar [a], Omprakash Kaiwartya[b], Upasana Dohare[a], Neeraj Kumar[c], Jaime Lloret[d]

[a] School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India.
(e-mail: skdohare@mail.jnu.ac.in, vipinjnu2010@gmail.com, upasanadohare@yahoo.com).
[b] School of Science & Technology, Nottingham Trent University, NG11 8NS, UK
(e-mail: omprakash.kaiwartya@ntu.ac.uk)
[c] Department of Computer Science and Engineering, Thapar Institute of Engineering, India,
(Email: neeraj.kumar@thapar.edu)
[d] Integrated Management Coastal Research Institue, Universitat Politecnica de Valencia, Camino de Vera 46022, Valencia, Spain (Email: jlloret@dcom.upv.es)

Abstract

Green communication in wireless sensor networks (WSNs) has witnessed significant attention due to the growing significance of sensor enabled smart environments. Energy optimization and communication optimization are two major themes of investigation for green communication. Due to the growing sensor density in smart environment, intelligently finding shortest path for green communication has been proven an NP-complete problem. Literature in green communication majorly focuses towards finding centralized optimal path solution. These centralized optimal-path finding solutions were suitable for application specific traditional WSNs environments. The cutting edge sensor enabled smart environments supporting heterogender applications require distributed optimal path finding solutions for green communication. In this context, this paper proposes a genetic algorithm enabled distributed zone approach for green communication. Specifically, instead of searching the optimal path solution in the whole network, the proposed algorithm identifies path in a small search space called distributed forward zone. The concept of forward zone enhances the searching convergence speed and reduces the computation centric communication cost. To encode the distributed routing solutions, variable length chromosomes are considered focusing on the target distributed area. The genetic algorithm enabled distributed zone approach prevents all the possibilities of forming the infeasible chromosomes. Crossover and truncation selection together generate a distributed path finding solution. To validate the experimental results with analytical results, various mathematical models for connectivity probability, expected end-to-end delay, expected energy consumption, and expected computational cost have been derived. The simulation results show that the proposed approach gives the high-quality solutions in comparison to the state-of-the-art techniques including Dijkstra's algorithm, compass routing, most forward within radius, Ahn-Ramakrishna's algorithm and reliable routing with distributed learning automaton (RRDLA).

Keywords

Wireless sensor networks, distributed path searching, optimization, Genetic algorithm, forward zone.

1. Introduction

Recent studies show that wireless sensor networks (WSNs) are used widely in several applications including battle field monitoring, forest fire detection, monitoring disaster areas, greenhouses, and environment monitoring [1, 2]. WSNs consist of a large number of sensors used to cover entire sensing region. These sensors may be thousands in numbers with short radio range, low processing capacity, limited memory and battery power [3]. These sensor nodes are densely deployed in the monitoring area, and information sensed by the sensors is transmitted to a destination point.

Because of short transmission range, communication between two neighbor sensors is possible only across a short distance. Therefore, to deliver the sensed data from any sensor to destination, intermediate sensors collaborate in further forwarding [4, 5]. Military surveillance is one of the WSNs applications where sensors are deployed in an area without any fixed infrastructure and some phenomenon is to be monitored. When the sensors detect the event, it is reported to the base station, which then takes appropriate action. Depending on the application, deployment of sensors may be either deterministic or random [6, 7]. Generally, in a non-hazardous area, sensors are deployed deterministically while in a hazardous area such as battlefield surveillance, a random deployment is preferred. In general, more number of sensors are required if the deployment is in random fashion as compared to deterministic placement [7]. In last few years, position-based routing is studied and extensively used in many applications. In position-based routing forwarding decisions are taken based on location information of sensors [8]. In many applications, energy consumption, and end-to-end delay along the route are important factors. Therefore, in a multi-hop network, one of the crucial issues is routing that has a major impact on performance of network [9].

To provide quality of service such as end-to- end delay for critical data, design of efficient sensor networks opens a new dimension of research. It is technically a challenging and complex problem to deliver information timely over network. Therefore, shortest path problem attracts interests of many researchers. Depending on applications, in routing, forwarding schemes can be based on minimizing end-to-end delay, energy consumption, number of hops, and geographic distance [10]. In order to support time-constrained services, such as military application, to deliver information within a specified time, an efficient routing algorithm is desired that should attempt to find shortest path quickly. The shortest path problem can be formulated as to find a path with minimal cost, from a set of multiple paths containing elected source and destination. In other words, the shortest path problem involves a classical combinatorial optimization problem which is NP-complete [9]. To solve such complicated problems, various evolutionary algorithms [11] have been proposed. One of the evolutionary algorithms is genetic algorithms (GA) [12]-[15] which is inspired by natural evolution and ensure solutions to optimization problems.

In this paper, we focus on determining the shortest route in a set of multiple routes with a minimum end-to-end delay requirement using heuristic or metaheuristic GA. Aim of the proposed GA-based routing is to solve the above described NP-complete optimization problem [14]. Many of the researchers have applied GA for whole networks and suggested an optimal solution at the cost of excessive computations. In particular, different form the existing works in [9]-[15], this paper proposes a sub-network called forward zone that not only covers a sufficient number of sensors to find the best route but also meets the quality of service requirements. Instead of applying GA on the whole network, GA is applied to a forward zone to reduce the computation cost and enhance path finding capacity. The main contributions of this paper are summarized as follows:

1) We formulate a problem of searching a shortest route which jointly minimizes energy consumption and communication delay. Such kind of optimization problem is an NP-complete problem.
2) To optimize energy consumption and delay of all the routes, we used GA that has good performance and less complexity. The major components of GA such as chromosome, selection, crossover and mutation operators have been mapped into the proposed optimization problem. A new method of loop free chromosome representation of the routes has been presented.
3) A concept of distributed forward zone is proposed to find the best route in small partition of the network that reduces the computation cost.
4) The objective functions: energy consumption and communication delay of the route have analytically derived. Analytical derivation of computation cost of the proposed solution is also established.
5) We implemented some of the existing routing algorithms i.e., Dijkstra's algorithm [16, 17],

compass routing (DIR) [18 - 20], most forward within radius (MFR) [20-21], Ahn-Ramakrishna's algorithm [9] and RRDLA [13] to compare the performance of the proposed routing algorithm.

The rest of the paper is organized as follows. In section 2, the related work is discussed. Section 3 presents design of the proposed GA based routing algorithm. In section 4, analytical framework, i.e., mathematical analysis of the proposed algorithm is presented. In section 5, simulation results of the proposed routing algorithm are compared with the existing routing algorithms. Finally, section 6 concludes this paper with a summary of the results and discussion.

2. Related work

2.1 Review of GA-based routing algorithms

In recent years, the shortest path problem has been investigated comprehensively in the literature [9 -14] and evolutionary algorithms draw attention to many researchers to find solution to the problem. Recent studies show that researchers have done some research work to solve the shortest path problem by applying genetic algorithms [9-15], particle swarm optimization [22], and neural networks [23]. To solve the shortest path problem, several routing algorithms such as Dijkstra's algorithm, Bellman-Ford algorithm, breadth-first algorithm, and depth-first algorithm have been proposed, to name a few [24]. These algorithms are suitable for fixed infrastructure networks since they give solution to the shortest path problem in polynomial time [9]. However, in an environment where network topology changes quickly, these algorithms show high computational complexity. Therefore, the indicated algorithms are not effective to satisfy the real-time communication [9].
Investigators have applied GA to give solution for various combinatorial optimization problems such as dynamic routing problem [13, 25] and multicasting routing problem [26]. In paper [27], Munemoto has been proposed a genetic algorithm where chromosomes are taken of variable length for encoding the problem. In mutation phase, a random gene or mutation node is selected from the chromosome. Another gene that is connected directly to that mutation node is selected randomly from the chromosomes. After that, by combining each partial chromosome, a mutated chromosome is produced. One partial chromosome represents a path from source to a selected node, and the other partial chromosome refers to a path from the selected node to the destination node. Inagaki [12] proposed a solution to routing problem that considers fixed length of chromosomes. In the Inagaki's algorithm, each gene represents a node ID, and the chromosomes are sequences of integers (node ID).
In paper [9], to solve the shortest path problem, a genetic algorithm has been presented. To encode the solution, variable length chromosomes have been taken on which crossover and mutation operations are performed. Also, a repair function to cure infeasible chromosomes and a population-sizing equation for determining a suitable population size has been developed. In [13], to solve the dynamic shortest path problem, genetic algorithms with immigrants and memory schemes have been proposed. This problem occurs because of changes in network topology over time due to node mobility in mobile wireless networks.

2.2 Review of greedy routing algorithms

In the past, some greedy based routing algorithms have been developed. In these algorithms, a sensor node takes forwarding decisions using location of itself, neighbors, and destination. Greedy routing algorithms can be based on progress, distance and direction [6]. Based on the notion of progress, Takagi and Kleinrock [21] presented a routing algorithm, called MFR. In this method, the neighbor with maximum progress is selected as next forwarder. Hou and Li [28] proposed nearest with forward progress (NFP) routing algorithm where each node sends packets to a nearest neighbor with forward progress. Distance based routing schemes are based on Euclidean distance between destination and neighbors of sender. In direction based routing next forwarder is selected based on deviation from line connecting sender and the destination [6]. Kranakis defined DIR [18], in which source or intermediate node selects a neighbor that is closest to the straight line between sender and the destination. GEDIR [19] algorithm is a greedy algorithm that always selects next forwarder that has minimum distance to the destination node. In most cases, it is found that the algorithm GEDIR finds same path to the destination as that of MFR [29].

The motivation behind the use of zone is that less information is needed in routing decision. Various zone based routing schemes such as location aided routing (LAR), and range DIRectional(R-DIR) are discussed in [20]. In paper [30], the concept of zone is introduced where each node has its routing zone separately. The zone radius of a node is defined as distance of the neighbors from a node. By using this radius, a routing zone is created which includes the neighbors having distance at most zone radius. In paper [31], author proposed RRDLA scheme for multi-constrained optimal path problem using distributed learning automaton (DLA) to find the smallest number of nodes. End-to-end reliability and delay are used in path selection process. Performance is measured in terms of end-to-end delay and energy-efficiency. In paper [32], authors investigated the problem of energy consumption in WSN. Authors proposed a routing metric including the residual energy, link quality, end-to-end delay, and distance. In [33], authors proposed OPEH for EH-WSNs considering the impact of the dynamic and heterogeneous duty cycle to reduce the end-to-end delay. In paper [34], author introduced advanced zonal stable election protocol (AZ-SEP) in which communication of sensor nodes with the base station is hybrid i.e., some nodes transmit the data directly to base station, while others use clustering mechanism.

Table I. Nomenclatures

Notation	**Description**	**Notation**	**Description**
N	Number of sensors	n	Number of node in forward area
R	Transmission range	r_k	k^{th} route
T_k	End-to-end delay of k^{th} route	FZ	Forward zone
E_k	Energy consumption of k^{th} route	R_z	Forward zone radius
$P_c(l)$,	Connectivity probability	C_i	A i^{th} chromosome (path)
$Ed(y)$	Expected distance	$E(F_i)$	Fitness function for energy consumption
$E(t_{cost})$.	Expected computational cost	E_{trans}	Transmission energy
$E(E_{total})$	Expected energy consumption	E_{rec}	Receiving energy
φ	Density of sensors	PSD	Probability of success delivery

3. Proposed GA for Shortest Path Problem

This section describes design of the proposed GA based routing algorithm. Procedure begins with organization of sensor deployment including location of all sensors and destination. Then, a specific region called forward zone is chosen, which can be identified by the coordinates in the network. Various components namely, genetic representation, initial population, fitness functions, selection operation, and crossover are involved in designing of the proposed algorithm. Source, intermediate and destination nodes make a routing path. Thus, genetic representation of any possible solution i.e., routing path from the source to the destination is represented in form of a chromosome. The proposed GA finds the optimal solutions in forward zone. If the desired solution is not obtained, size of forward zone is expanded. The process is repeated until the desired solution obtained. We used shortest path and optimal path interchangeably in this paper. The symbols used in this paper are described in table-I. The several components of GA have been discussed in further sections.

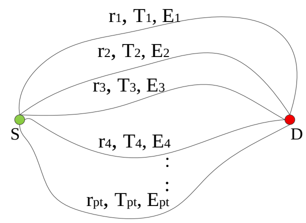


Fig. 1.Multiple routes from Source S to Destination D

3.1 *Problem formulation*

We consider a two-dimensional sensing field having N number of sensors. A large number of sensors are deployed randomly in WSN having same initial energy and transmission range R. It is assumed that the coordinates of all sensors are known. WSN is modeled as an undirected graph $G = (V, L)$ where, V is the set of sensor nodes and L is the set of links between sensors. A link between two nodes v_i and v_j exists iff $d_{i,j} \leq R$, where $d_{i,j}$ is the distance between nodes v_i *and* v_j. In WSN, there may be multiple paths from source S to destination D (cf. Fig.1), and prior to transmission of data packets, a routing path is required. Our objective is to minimize end-to-end delay, and energy consumption. To achieve the objective, an efficient approach is required to find an optimal path r_{opt} among a set of multiple paths $RP = \{r_1, r_2, \dots r_{pt}\}$ such that $r_{opt} = \{r_k \mid T_k = min\,(T_1, T_2, \dots T_{pt})\}$ and $r_{opt} = \{r_k \mid E_k = min\,(E_1, E_2, \dots E_{pt})\}$ where, pt is total number of routing paths from S to D, T_k and E_k are end-to-end delay and energy consumption along the k^{th} route, respectively.

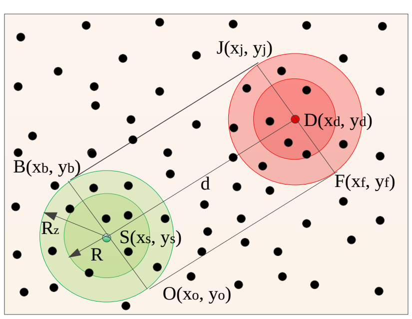


Fig. 2.Forward zone and coordinate assignment.

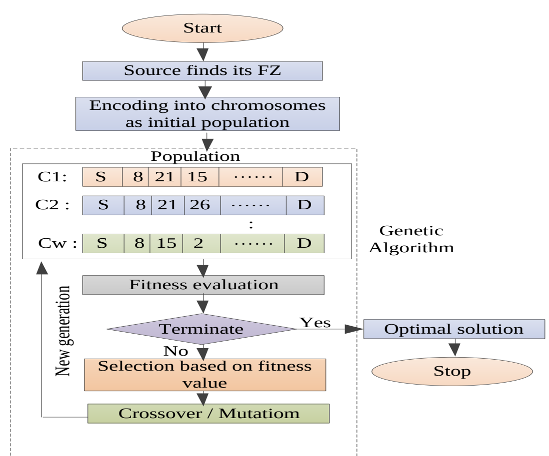


Fig. 3.Flow chart of proposed GA.

3.2 *Formulating routing as an optimization problem*

In this section, the problem of finding a shortest path for end-to-end delay, and energy consumption is presented. Note that based on WSNs applications, delay constraints, and energy constraints are bounded to thresholds T_{TH}, and E_{TH} respectively. For instance, delay sensitive applications require low value of T_{TH}. However, assigning high value for T_{TH} corresponds to delay-tolerant applications. The end-to-end delay of a route r_i defines time it takes for a packet to reach at the destination node. Based on the proposed algorithm, delay over route r_i is computed by $T_i = \sum_{i=1}^{e} T(v_i,\ v_{i+1})$, where e is number of edges along the route r_i, while $T(v_i,\ v_{i+1})$ denotes delay between two neighbor nodes v_i and v_{i+1}. The value of e may vary depending on the path length. Thus, the problem of finding shortest path to minimize end-to-end delay, and energy consumption can be formulated as an optimization problem and given by

$$\left.\begin{matrix} min\, T_k \\ min\, E_k \end{matrix}\right\} \quad (1)$$

Subject to

$$T_k = \sum_{i=1}^{e} T_i(v_i,\ v_{i+1}) \le T_{TH} \quad (2)$$

$$E_k = \sum_{i=1}^{e} E_i(v_i,\ v_{i+1}) \le E_{TH} \quad (3)$$

3.3 Formation of Forward zone

In this section, crucial part of the proposed algorithm is presented. To find a forward zone (FZ), two circles $Cr(S)$ and $Cr(D)$ with forward zone radius R_z are formed, where S and D are source and destination nodes respectively. Thereafter, two parallel tangents $Tn(B,J)$ and $Tn(O,F)$ are drawn from $Cr(S)$ to $Cr(D)$. The region $BOFJ$ made by tangents and two diameters BO, JF (lines that connect the points of contacts of tangents) is considered as FZ [cf. Fig.2]. The next half of the circle formed with radius R towards destination D is called forward area (FA) and the other half circle is termed as backward area (BA) of S. A FA of node consists of multi-hop neighbors. The coordinates of the sender and destination are $S(x_s, y_s)$ and $D(x_d, y_d)$, respectively, and the coordinates of the FZ are given as $B(x_b, y_b)$,$O(\mathrm{x_o}, \mathrm{y_o})$, $F(x_f, y_f)$, $J(x_j, y_j)$ (cf. 2).

Definition 1. *For any sensor node v_i, its transmission region is defined as the circle, denoted by $Cr(v_i)$, centered at position (x_i, y_i) of v_i with radius r_{v_i} where, $r_{v_i} = R$, where R is the transmission range of the sensor(cf. Fig. 2).*

Definition 2. *If $d_{i,j}$ is the distance between two nodes v_i and v_j, for sensor node v_i, its neighboring nodes are defined as $N_{nb}(v_i) = \{v_j | v_j \in V, d_{i,j} \le R\}$ where, $N_{nb}(v_i)$ denotes the set of neighbors of a sensor.*

Definition 3. *For a given destination node v_d, the neighbor nodes of v_i lie in forward direction are $N_f(v_i) = \{v_j | v_j \in N_{nb}(v_i), d_{j,d} \le d_{i,d}\}$. The neighbor nodes of v_i, lying in backward direction is defined as $N_b(v_i) = \{v_j | v_j \in N_{nb}(v_i), d_{j,d} > d_{i,d}\}$ where $d_{i,d}$ denotes the distance between v_i and v_d, and $d_{j,d}$ denotes the distance between neighbor node v_j and v_d.*

Definition 4. *Given a sensor node v_i with coordinates (x_i, y_i), it will belong to forward zone FZ iff (x_i, y_i) lies within the coordinates of FZ $\{B(x_b, y_b), O(x_o, y_o), J(x_j, y_j), F(x_f, y_f)\}$.*

Definition 5. *For a given sensor node v_i, if $N_f(v_i) \cap N(FZ) = 0$ i.e., no neighbor node exists in the forward area of v_i in FZ then size of the FZ is increased, so that $N_f(v_i) \cap N(FZ) \ne 0$. Here, $N(FZ)$ represents the nodes lying in FZ.*

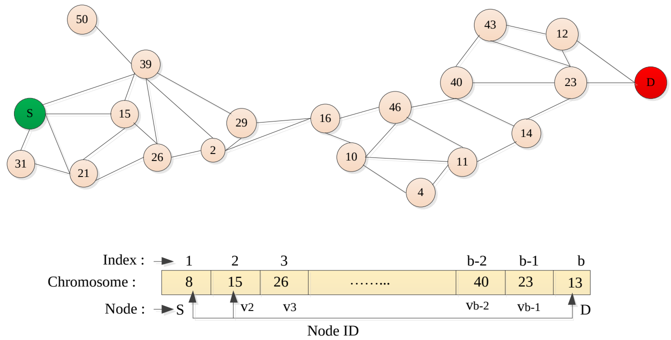


Fig. 4.Example of network and encoding scheme.

3.4 *Genetic representation and loop-free chromosome formation*

The flowchart in Fig. 3 represents the process of finding an optimal solution using GA which includes encoding scheme, population initialization, fitness function evaluation, selection, crossover and termination. In the proposed algorithm, it is considered that chromosomes which represent possible solutions are the sequence of genes where each gene represents a node ID [9 13]. The chromosomes have been taken of variable length. The maximum length of a chromosome is z where, z is the number of nodes in FZ. Fig. 4 depicts an example of encoding a route into a chromosome. It is a sequence of node IDs of the nodes participated in a route from source S to destination D. In Fig. 4, b is the number of nodes along the constructed path. Source node ID and destination node ID are always stored at the gene of first and last index respectively, and the rest of the genes can be arranged using procedure 2.

***Procedure 1: Find_Forwarder** (S, D, FZ, R, N, Dist)*

```
Input: FZ, S, D, R, N, Dist  /*Dist contains the distances between each node
Output: Forwarder_node
begin
    counter=1
    for (i=1 to N)
        if ((Dist[S, i] <=R) &(Dist (i, D) <Dist(S, D))&(i ∈ FZ))
            Forwarder_List [counter] = i    /* Add node i to the forwarding
                                               nodes list of a node*/
            counter=counter+1
        end if
    end for
    if(D∈Forwarder_List)
        Forwarder_node = D
    else
    Forwarder_node = rand (Forwarder_List)  /*Selection of a node
                                              randomly from Forwarder_List*/
    end if
    Output Forwarder_node
End
```

Fig. 5.Pseudo code of the forwarder selection procedure

In the chromosome formation process [35, 36] (cf. procedure. 2), initially, source node S is stored at first index of the chromosome. Then, a forwarder node, say v_i (cf. procedure. 1) from the neighbor set $N_f(S)$ of source node S is picked and stored at the next index of the chromosome that makes a partial route $S \rightarrow v_i$. Thereafter, a node v_j from $N_f(v_i)$ is picked (cf. procedure. 1) and added it into the partial route which is extended to $S \rightarrow v_i \rightarrow v_j$. This process is repeated until the destination node D is reached. In this way, a random chromosome (path) $C_{(S,\ D)} = \{S,\ v_i, v_j \ldots, D\}$ is formed. The chromosome formation process eliminates the lethal genes (creating a loop) by selecting the next forwarder node in forward direction. It prevents all the possibilities of

***Procedure 2: Chromosome_Form**(S, D, FZ, R, N, Dist)*

```
Input:  S, D, FZ, R, N, Dist
Output: Chromosome   // It contains the node id's
begin
    Initially j=1
    Chromosome[j]=S
    while(S! = D)
        C2= Find_Forwarder (S, D, FZ, R, N, Dist)
        j=j+1
        Chromosome[j]=C2
        S=C2
    end while
    Output Chromosome
end
```

Fig. 6.Pseudo code of the chromosome formation procedure.

forming the infeasible chromosomes [37]. An example of chromosome encoding is shown in Table I by considering a network shown in Fig. 4. Initially, the source node S is encoded as the gene of first index of the chromosome. From this figure, it can be seen that node S has three forwarding nodes i.e., 21, 15, and 39. A second gene i.e., 15 is picked randomly from the forwarding nodes list and stored it into second index of chromosome. Each time a new gene is added, a partial route (chromosome) is formed. In this way, all the intermediate genes are filled (cf. procedure. 2) until the destination is reached (the process of chromosome formation is show in the table-II).

3.5 Population Initialization

TABLE II
CHROMOSOME FORMATION

Sender node	Forwarding nodes list (genes)	Random node in forwarding nodes list	Partial route (Chromosome)								
S	{21, 15, 39}	15	S	15							
15	{26, 39}	26	S	15	26						
26	{2}	2	S	15	26	2					
2	{29, 16}	16	S	15	26	2	16				
16	{10, 46}	46	S	15	26	2	16	46			
46	{40, 11}	40	S	15	26	2	16	46	40		
40	{43, 23, 14}	23	S	15	26	2	16	46	40	23	
23	{D}	D	S	15	26	2	16	46	40	23	D

Initially, a number of chromosomes are generated randomly which make together initial population. Repeating the chromosome formation process for a number of times, say w, the initial population $IP = [C_1,\ C_2, C_3 \ldots\ldots C_w]$ can be obtained.

3.6 Fitness Functions

3.6.1 End-to-End delay

In GA, fitness is the crucial part. Purpose of the fitness function is to evaluate quality of solution accurately. In the proposed algorithm, two fitness functions are designed to reduce end-to-end delay, and minimize energy consumption, in a path between S and D. This Fitness function evaluates quality of chromosome in population and determines whether a chromosome reduces end-to-end delay or not. Therefore, among a set of multiple solutions, the one with a minimum end-to-end delay is chosen. The fitness function is defined as follows

$$T(F_i) = \sum_{j=1}^{b_i-1} T_{(v_i(j),\ v_i\,(j+1))} \tag{4}$$

where, F_i represents the fitness value of i^{th} chromosome, b_i is the length of i^{th} chromosome, and $v_i(j)$ represents a node of j^{th} index in i^{th} chromosome. $T_{(v_i(j),\ v_i\,(j+1))} = \frac{d_{(j,\ j+1)}}{c}$ is the propagation delay along a link between two nodes, where, $d_{(j,\ j+1)}$ is distance between neighbor nodes $v_i(j)$ and $v_i\,(j+1)$ and c is propagation speed.

3.6.2 Energy consumption

This fitness function evaluates the energy consumption for different routes. Therefore, among multiple solutions, the one with minimum fitness value is chosen. The fitness function for energy consumption is defined as follows

$$E(F_i) = \sum_{j=1}^{b_i-1} E_{(v_i(j),\ v_i(j+1))} \tag{5}$$

where, $E_{(v_i(j),\ v_i(j+1))} = \left(2 \times Elec + Amp \times (d_{(j,\ j+1)})^{\emptyset}\right) \times k$ is the energy consumption in transmission and reception of a data packet of size k-bit, from node $v_i(j)$ to $v_i\ (j+1)$. A simple well known first order radio model [38] is used to measure the energy consumption. According to this model, transmission energy E_{trans} and receiving energy E_{rec} can be defined as $E_{trans} = \left(Elec + Amp \times (d_{(j,\ j+1)})^{\emptyset}\right) \times k$ and $E_{rec} = Elec \times k$ respectively, where Ø is path loss coefficient, $d_{(j,\ j+1)}$ is the distance between two nodes, $Elec$ is energy dissipated to run the transmitter or receiver circuitry and Amp is energy required for transmit amplifier. The value of Ø normally lies in the range of 2 to 4, depending on the environments such as free space and lossy environments [39]. In this work, we consider free space environment i.e., Ø = 2. Therefore, total energy E_t consumed to deliver k-bits data from source to destination can be expressed as

$$E_t = (E_{trans} + E_{rec}) \times H_c$$
$$= \left(2 \times Elec + Amp \times (d_{(j,\ j+1)})^{\emptyset}\right) \times k \times H_c,$$

Where, H_c is the number of links along the route from source S to destination D.

3.7 Selection

The purpose of the selection operation is to improve the quality of population. It must be noted that selection operator plays a crucial role to produce high quality of solution. Hence, selection operator must be efficient enough to converge the solution quickly. Two basic type of selection methods are proportionate and ordinal-based selection [9]. Proportionate selection selects chromosomes on the basis of their relative fitness values, and ordinal-based selections pick out chromosomes based on their rank instead of fitness value. In our algorithm, truncation selection [15] is adopted, where high quality chromosomes of current population get a chance by copying them into the next generation.

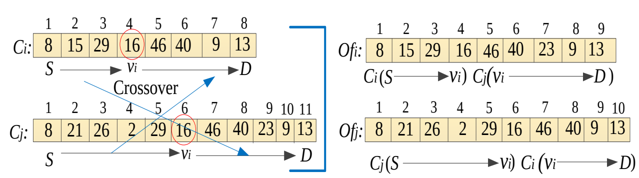


Fig. 7.Crossover operation and two new offspring

3.8 Crossover and Mutation

Crossover and mutation are two basic operators used in GA. Crossover is one of the important parts of the genetic algorithm components. In the crossover operation, each time two chromosomes C_i and C_j are selected from current population so as to produce better ones. In the proposed algorithm, single-point crossover [9 14] is adopted to exchange partial chromosomes where both C_i and C_j should retain at least one common gene. From the set of common genes, one gene, say v_i, is randomly selected. Now, $C_i = [S \to v_1 \to v_2 \ldots . \to D]$ can be represented as a combination of two partial chromosomes $C_i\{S \to v_i\}$ and $C_i\{v_i \to D\}$. Similarly, $C_j = [S \to v_1 \to v_2 \to v_3 \ldots . \to D]$ can be represented as a combination of two partial chromosomes $C_j\{S \to v_i\}$ and $C_j\{v_i \to D\}$. The crossover operation exchanges $C_i\{v_i \to D\}$ and $C_j\{v_i \to D\}$ and two new offspring $Of_i[C_i\{S \to v_i\} \to C_j\{v_i \to D\}]$ and $Of_j[C_j\{S \to v_i\} \to C_i\{v_i \to D\}]$ are formed (cf. Fig.7). The procedure of crossover operation is expressed in Fig. 8. After executing the crossover operation, the mutation operator is applied to the population to maintain population diversity. With mutation probability, in a chromosome, one common gene is randomly picked as a mutation node, and a partial route from mutation point to the destination is generated. Mutation allows the GA to avoid local optima [9 13 14].

3.9 *Termination*

Procedure3: Crossover operation

```
-----------------------------------------------------------------------------------
Input: Chromosomes C, C1          // Input chromosomes
Output: Ch, Ch1                   // Output new Childs
begin
1. Crossover (C, C1)
2.  LC=length(C)                  //Length of chromosome C
3.  LC1=length (C1)               //Length of chromosome C1
4.for (i=1 to LC)
5.     for (j=1 to LC1)
6.          if (C(i) = = C1(j))
7.               CS= (i, j)               // Indexes of common genes
8.          end if
9.     end for
10.end for
11.s(1, 2)= random(CS)     // randomly selection of any pair from CS
12.for (i=1to s(1))
13.Ch(i)=C(i)
14.end for
15.for ( i=1 to LC1-s(2))
16.Ch(s(1)+i)=C1(s(2)+i)    // Ist exchange of partial chromosome
17.end for
18.for ( i=1 to s(2))
19.Ch1(i)=C1(i)
20.end for
21.for (i=1 to LC-s(1))
22.     Ch1(s(2)+i)=C(s(1)+i)   // IInd exchange of partial chromosome
23.end for
24.Output Ch, Ch1
end
```

Fig. 8.Pseudo code of the crossover operation

Termination is a criterion that decides end of searching. After each generation, the criterion is tested. In our implementation, the maximum number of generations is used as termination criteria. Other than the maximum number of generations, termination criteria can be based on fitness threshold, maximum computing time, fitness convergence, and population convergence.

4. Analytical Framework

TABLE III
PARAMETER SETTINGS

Parameter	Value
Area (A)	500 m×500 m
Network size (N)	50-200 nodes
Transmission range (R)	60-100 m
Forward zone radius (R_z)	60-100 m
Population size (pop)	50-150
$Generations$	1-14
Crossover probability	0.7
Mutation probability	0.3
$Elec$	50nJ/bit
Amp	100pJ/bit/m^2
Path loss coefficient ($\emptyset$)	2
Data packet size (k)	512 bit

In this section, we derived analytical expressions for connectivity probability $P_c(l)$, expected distance $Ed(y)$ between sender and next forwarder, expected distance $Ed(x)$ between source and destination, expected end-to-end delay $EEtE(T_e)$, expected energy consumption $E(E_{total})$, and expected computational cost $E(t_{cost})$. The parameters values used in our experiments are listed in Table III.

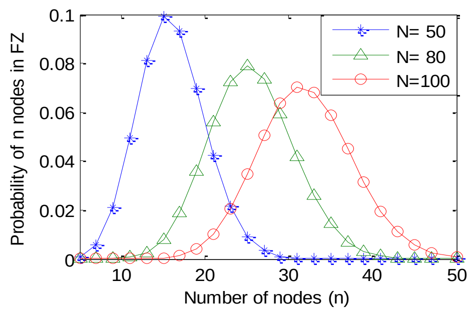


Fig. 9.Probability of number of nodes present in FZ with different network size

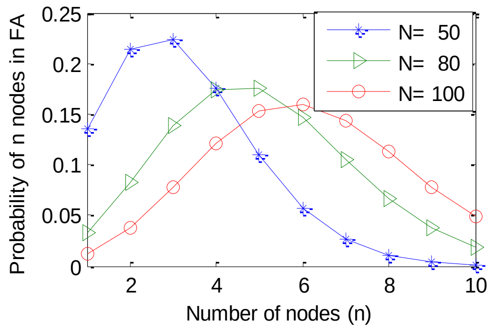


Fig.10. Probability of number of nodes present in FA with different network size

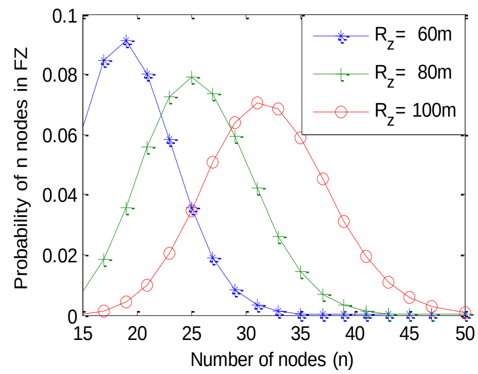


Fig.11. Probability of number of nodes in FZ with different transmission range

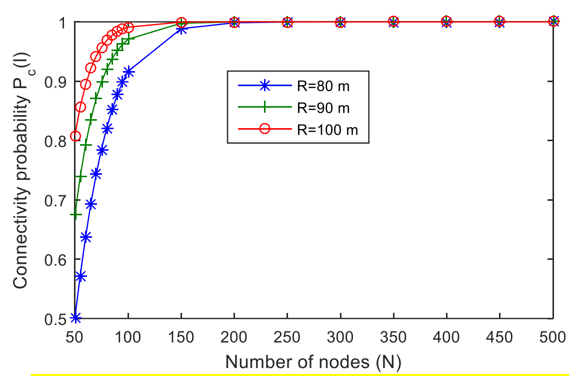


Fig.12. Connectivity probability changes with both transmission range and network size.

It is assumed that the area of entire sensing region is $A\ m^2$ and the sensing region is assumed to be a square with one of the sides of $L\ m$. The node density φ can be expressed as $\varphi = \frac{N}{A}\ nodes/m^2$. Let the distance between source node S and destination node D is d. The area of FZ and forward area (FA) can be expressed as $A_{FZ} = 2R_z d$ and $A_{FA} = \pi R^2/2$, respectively. It is assumed that the sensor nodes are distributed using Poisson distribution [40] with random variable Z. The probability of n nodes present in FZ is given by

$$PA_{FZ}(Z = n) = \frac{(\varphi A_{FZ})^n \cdot e^{-\varphi A_{FZ}}}{n!} = \frac{(\varphi 2R_z d)^n \cdot e^{-\varphi 2R_z d}}{n!} \tag{6}$$

Similarly, for a random variable Y that represents the number of sensor nodes in FA, the probability of n number of nodes present in the FA is given by

$$PA_{FA}(Y = n) = \frac{(\varphi A_{FA})^n \cdot e^{-\varphi A_{FA}}}{n!} = \frac{(\varphi \pi R^2/2)^n \cdot e^{-\varphi \pi R^2/2}}{n!} \tag{7}$$

4.1 Connectivity probability $P_c(l)$

Let $P_c(l)$ is the probability of connectivity of a route of length H_c. A link l_i exists if there are two consecutive nodes and the distance between themlies within the transmission range. In other words, for link connectivity, at least one node must exist in FA. The probability $P_c(l_i)$ of at least one link exists between any two nodes in FA can be expressed as

$$P_c(l_i) = 1 - e^{-\varphi\pi R^2/2}$$

Now, connectivity probability of a route can be expressed as

$$P_c(l) = \prod_{i=1}^{H_c} P_c(l_i) = \prod_{i=1}^{H_c}(1 - e^{-\varphi\pi R^2/2}) \ = \left(1 - e^{-\varphi\pi R^2/2}\right)^{H_c} \tag{8}$$

Fig. 9 and 10 show the probability of number of nodes present in FZ and FA, respectively for different values of N. From these figures, it can be seen that when the value of N increases, the occurrence of nodes in FZ and FA also increases. These results are obtained for the network area of 500 m×500 m, and zone radius of 100 m. For N=50, the highest value of number of nodes in FZ is 15 and for N=80 and 100, the highest number of nodes lying in FZ are 25 and 31, respectively. Similarly, Fig.10 depicts that the probability of number of nodes 3, 5, and 6 present in FA is high for the values of N= 50, 80, and 100, respectively. This is because, when N increases, the node density increases in the same area. Therefore, as expected, the probability of number of nodes occurring in FZ and FA is increasing.

Fig.11 shows variation in the value of n with the different value of R_z. It represents the results obtained for N=100. It isnoticed that for the values of R_z=60 m, 80 m, and 100 m, the probability of nodes occurring in FZ is high for the values of n=19, 25, and 31, respectively. When R_z increases, area of FZ also increases because the area of FZ depends on the value of R_z. Consequently, the number of nodes in FZ is increasing.

Fig. 12 shows the variation in the connectivity probability, with different values of R. Experiments are conducted for the network size of 50-500 nodes. It is noted that as the value of N increases, the connectivity probability is increasing. When transmission range R is 80 m, the highest probability is obtained for network size of 200-500. For R=90 m and 100 m, the connectivity probability is close to 1, when the values of N are 140 and 110, respectively. The reason is that when the value of N increases, there are more chances of falling more number of nodes in FA. Therefore, the probability of at least one node present in FA increases with the increasing in network size. So, to get the proper connectivity, the value of N and R can be adjusted as per the requirements of the applications.

4.2 Probability of success delivery (PSD)

Let P_{rs} is the instantaneous power of received signal, I is the interfering power, and Ω is the minimum required power to detect the received signal. Probability of success delivery for a link between v_i and v_j is given by

$$PSD_{i,j} = P\left(\frac{P_{rs}}{I} > \Omega\right) \tag{9}$$

Probability of success delivery for a route of length H_c can be calculated as

$$PSD = (PSD_{i,j})^{H_c} \tag{10}$$

4.3 Expected distance $Ed(y)$ between source and next forwarder

It is assumed that a source sensor S has n number of neighbors in FA. Let N_{fd} is the next best forwarder sensor which has distance y from S. Let $y_1, y_2, y_3, \dots, y_n$ denotes the distance between source and its neighbors exist in FA, respectively. The expected value of y can be calculated as follows. Let $F(y)$ and $f(y)$ are cumulative density function (CDF) and probability density function (PDF) of y respectively. Then, $F(y)$ can be expressed as

$F(y) = (\frac{y}{R})^n$, where R is the transmission range.

By using $F(y)$, the value of $f(y)$ can be expressed as

$$f(y) = \frac{d}{dy}F(y) = \frac{1}{R^n}ny^{(n-1)} = \frac{n}{R}(\frac{y}{R})^{(n-1)}$$

$$Ed(y) = \int_0^R yf(y)dy = \frac{n}{R^n}\int_0^R y^n dy = \frac{n}{R^n}[\frac{R^{(n+1)}}{n+1}]$$

$$= \frac{n}{(n+1)}R \tag{11}$$

4.4 *Expected distance $Ed(x)$ between source and destination*

In our work, it is assumed that the sensing region is square in shape. Then, the maximum distance between source S and destination D can be $L\sqrt{2}$, if S and D are situated at ends of the diagonal. Let the expected distance $Ed(x)$ between S and D is x. The CDF $F(x)$ of x can be expressed as $\frac{x}{L\sqrt{2}}$ and PDF $f(x)$ of x can be defined as

$$f(x) = \frac{d}{dx}F(x) = \frac{1}{L\sqrt{2}}$$

$Ed(x) = \int_{\omega}^{L\sqrt{2}} xf(x)dx = \frac{1}{2L\sqrt{2}}\left[(L\sqrt{2})^2 - \omega^2\right]$, where ω is lower limit. (12)

For simplicity, it is assumed that $\omega = 1$, because it is very small value. Now, (12) can be written as

$$Ed(x) = \frac{1}{2L\sqrt{2}}\left[(L\sqrt{2})^2 - 1\right] \tag{13}$$

4.5 *Expected End-to-End delay $EEtE(T_e)$*

In this section, we evaluate the expected end-to-end delay along the routing path. To evaluate the delay, here, only propagation delay is considered. A routing path passes through various sensor nodes, is the set of connected links. So first, we calculate the expected delay of a link between two nodes. After that, expected end-to-end delay is derived. Let l is the length of a link between two nodes v_i and v_j. Then propagation delay t_p is given by $t_p = \frac{l}{c}$, where c is the propagation speed. Now the PDF of t_p can be calculated as

$f(t_p) = K \times \frac{l}{c}$, where K is a constant (14)

$\int_{\sigma}^{R} K \times \frac{l}{c} dl = 1$, where σ is lower limit and R is transmission range.

$$K = \frac{2c}{(R^2 - \sigma^2)}$$

By putting the value of K in (14), the PDF $f(t_p)$ can be written as

$$f(t_p) = \frac{2l}{(R^2 - \sigma^2)}$$

The expected propagation delay $Ep(t_p)$ of a link can be calculated as

$$Ep(t_p) = \int_{\sigma}^{R} \frac{l}{c} \times f(t_p)\, dl$$

$$= \int_{\sigma}^{R} \frac{l}{c} \times \frac{2l}{(R^2 - \sigma^2)} dl = \frac{2}{3c(R^2 - \sigma^2)}[R^3 - \sigma^3]$$

$$= \frac{2}{3c}\frac{(R^2 + R\sigma + \sigma^2)}{(R + \sigma)} \tag{15}$$

For simplicity, it is assumed that $\sigma = 1$, Now, (15) can be represented as

$$Ep(t_p) = \frac{2}{3c}\frac{(R^2 + R + 1)}{(R + 1)} \tag{16}$$

The expected end-to-end delay of a route can be expressed as

$$EEtE(T_e) = E(t_p) \times H_c = \frac{2}{3c}\frac{(R^2 + R + 1)}{(R + 1)} \times H_c\ , \tag{17}$$

Where, H_c is route length and can be expressed as $H_c = \frac{Ed(x)}{Ed(y)}$.

4.6 *Expected energy consumption* $E(E_{total})$

In this section, we evaluate the expected energy consumption in delivery of data along the routing path. To evaluate the energy consumption, here only transmission and reception energy are considered. First, we calculated the expected energy required in the transmission and reception of data packet between two neighbor nodes [8]. After that, the expected energy consumption $E(E_{total})$ in forwarding of a data packet from source S to destination D is derived. In the delivery of k-bit data through a link of length l between two nodes v_i and v_j, the energy consumption E_t is defined as

$$E_t = (2k \times Elec + k \times Amp \times l^2) \quad (18)$$

The PDF $f(E_t)$ of the energy consumption E_t can be calculated as

$$f(E_t) = K \times (2k \times Elec + k \times Amp \times l^2)$$
$$= K \times (a_1 + b_1 \times l^2) \quad (19)$$

where, $a_1 = 2k \times Elec$ and $b_1 = k \times Amp$

$\int_\sigma^R K \times (a_1 + b_1 \times l^2)\, dl = 1$, where σ is lower limit and R is transmission range.

$$K = \frac{1}{\left(a_1 R + b_1 \frac{R^3}{3}\right) - \left(a_1 \sigma + b_1 \frac{\sigma^3}{3}\right)}$$

Therefore, $f(E_t)$ can be expressed as

$$f(E_t) = \frac{1}{\left(a_1 R + b_1 \frac{R^3}{3}\right) - \left(a_1 \sigma + b_1 \frac{\sigma^3}{3}\right)} \times (a_1 + b_1 \times l^2)$$

The expected energy consumption $E(E_t)$ can be calculated as

$$E(E_t) = \int_\sigma^R (a_1 + b_1 \times l^2) \times f(E_t)\, dl$$
$$= \int_\sigma^R \frac{(a_1 + b_1 \times l^2)^2}{\left(a_1 R + b_1 \frac{R^3}{3}\right) - \left(a_1 \sigma + b_1 \frac{\sigma^3}{3}\right)}\, dl$$
$$= c_1 \left[a_1{}^2 l + b_1{}^2 \frac{l^5}{5} + 2a_1 b_1 \frac{l^3}{3}\right]_\sigma^R$$
$$= c_1 \left[\left(a_1{}^2 R + b_1{}^2 \frac{R^5}{5} + 2a_1 b_1 \frac{R^3}{3}\right) - \left(a_1{}^2 \sigma + b_1{}^2 \frac{\sigma^5}{5} + 2a_1 b_1 \frac{\sigma^3}{3}\right)\right] \quad (20)$$

For simplicity, it is assumed that $\sigma = 1$, because it is very small value. The expected energy consumption can be expressed as

$$E(E_t) = c_1 \left[\left(a_1{}^2 R + b_1{}^2 \frac{R^5}{5} + 2a_1 b_1 \frac{R^3}{3}\right) - \left(a_1{}^2 + \frac{1}{5} b_1{}^2 + \frac{2}{3} a_1 b_1\right)\right],$$

where, $c_1 = \frac{1}{\left(a_1 R + b_1 \frac{R^3}{3}\right) - \left(a_1 + b_1 * \frac{1}{3}\right)}$

In the delivery of k-bits data through a routing path from source to destination nodes, $E(E_{total})$ can be expressed as

$$E(E_{total}) = E(E_t) \times H_c$$
$$= c_1 \left[\left(a_1{}^2 R + b_1{}^2 \frac{R^5}{5} + 2a_1 b_1 \frac{R^3}{3}\right) - \left(a_1{}^2 + \frac{1}{5} b_1{}^2 + \frac{2}{3} a_1 b_1\right)\right] \times H_c \quad (21)$$

4.7 Expected computation cost $E(t_{cost})$

In this section, we derive the mathematical expression for expected computation cost required in searching the shortest route. For simplicity, here, it is assumed that entire sensing region forms an n-tree type of topology (cf., Fig.15), where each node has only n number of nodes in forward direction and each leaf node is connected to destination node D.

The number of paths from a source to the one-hop neighbor nodes is given by $n = \varphi A_{FA} = \varphi\pi R^2/2$. Now, total number of paths from the source to the neighbors of two-hop length can be expressed as $n \times n$. Accordingly, the total number of paths from the source to the destination in the entire region is calculated as

$$T_{path} = \prod_{i=1}^{H_c-1} n = n^{(H_c-1)} \tag{22}$$

Let the computation cost of a path is δ. The total computation cost T_{cost} for searching an optimal

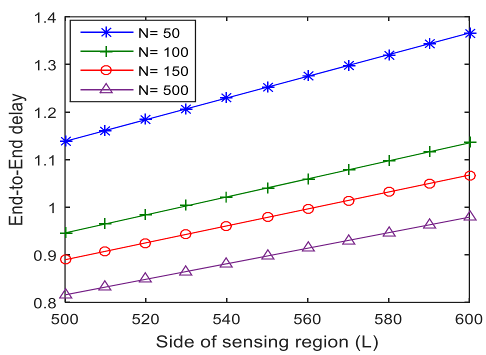


Fig.13.Expected End-to-End delay changes with both size of region and network size

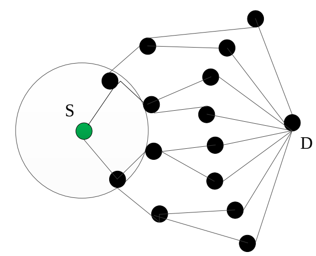


Fig.15. An example of n-tree topology.

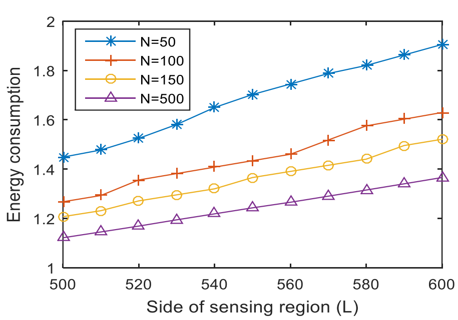


Fig.14.Expected Energy consumption changes with both size of region and network size

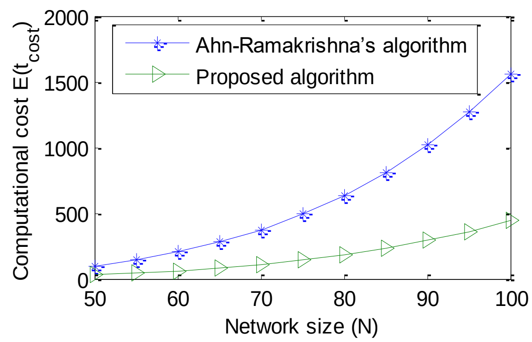


Fig.16. Comparison results of computation cost.

path in entire region can be defined as $T_{cost} = T_{path} \times \delta$. The path density denoted by ρ , can be represented as $\rho = T_{path}/A$. Then, number of expected paths from the source to the destination in FZ can be easily calculated using Poisson distribution. If M is a random variable that represents the number of paths in FZ, then the probability of ξ paths present in FZ is given by

$$PA_{FZ}(M = \xi) = \frac{(\rho A_{FZ})^{\xi} \cdot e^{-\rho A_{FZ}}}{\xi!}$$
$$= \frac{(\rho 2R_z d)^{\xi} \cdot e^{-\rho 2R_z d}}{\xi!} \tag{23}$$

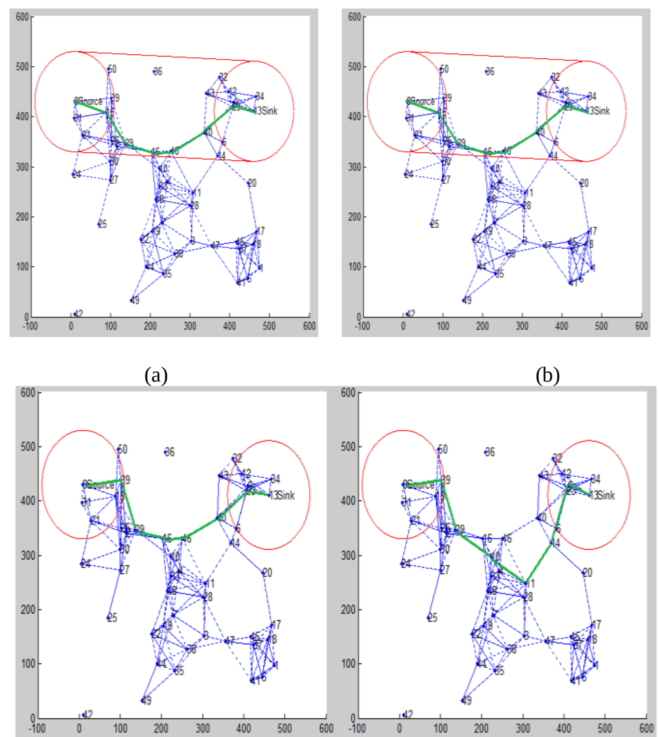
By using the Poisson distribution property, the expected number of path in FZ will be $E(t_{path}) = \rho A_{FZ}$. Therefore, to find the optimal path, when forward zone is considered, the expected computation cost $E(t_{cost})$ can be expressed as

$$E(t_{cost}) = E(t_{path}) \times \delta = \rho A_{FZ} \times \delta \tag{24}$$

In order to further compare the performance of the proposed genetic algorithm with that of Ahn-Ramakrishna's algorithm, the computation cost obtained through simulations is presented in Fig. 16. The average computation cost of the proposed algorithm is 176.8048 units, and that of the Ahn-Ramakrishna's algorithm is 625.1004 units. In our case, the computation cost does not increase significantly with the number of nodes in the network while it does in the case of Ahn-Ramakrishna's algorithm. It is noted that as the network size increases, the computation cost for both the algorithms increases.
From Fig. 16, it can be observed that for network size 50, the computation cost for the proposed GA and Ahn-Ramakrishna's algorithm is 27.49 units and 97.21 units, respectively. The reason of lower computation cost in the proposed GA is that it attempts to find the shortest path, only in a substantial region called as FZ. Similarly, for network size N=100, the computation cost for the proposed GA and Ahn-Ramakrishna's algorithm increases to 439.93 units and 1555.38 units, respectively. It can be observed that the proposed GA performs better in the large network as compared to Ahn-Ramakrishna's algorithm.

5. Experiments and Discussion

In this section, to evaluate the performance of the proposed algorithm, a series of experiments are conducted by using MATLAB tool. To measure the performance of our algorithm, following method is used to generate the initial network topology. Primarily, a square region with the area of 500 m×500 m is specified. This area has the height [0-500] on y-axis and the width [0-500] on x-axis. Then within the square region, 50-100 nodes are generated randomly where each node is specified with the position (x, y).It is assumed that each sensor knows its position by using global positioning system or other positioning systems. In this random network topology, a link between two nodes is added if the distance between them lies within the radio transmission range R. Finally, it is checked that if the generated topology is connected or not, if not, the previous process is applied repeatedly until a connected topology is found. To generate the connected topology, either the number of nodes or the transmission range may be increased. The parameters values used in our experiments are listed
in Table III.



(a) (b) (c) (d)

Fig.17. Sensors deployment and comparison results for the shortest path found by the algorithms. (a) The result of proposed algorithm (End-to-End delay: 1.7060 µs). (b) The result of Dijkstra's algorithm (End-to-End delay: 1.7060 µs). (c) The result of DIR algorithm (End-to-End delay: 1.8049 µs). (d) The result of MFR algorithm (End-to-End delay: 2.1632µs.

To perform the simulation experiments, various sizes of networks have been taken randomly. The performance of the proposed algorithm and comparison with the various routing algorithms: Dijkstra's, DIR and MFR, are performed. In the experiments, the source node and the destination node are node numbers 8 and 13 respectively (cf. Fig.17).

5.1 Simulation results for Network with 50 nodes

Fig. 17(a)-(d) show the comparison of shortest paths (bold green lines) computed by the algorithms. Through simulation, it is seen that as compared to the existing algorithms, the proposed algorithm gives the better results in terms of the shortest path with minimum end-to-end delay. Here, it can be observed that the obtained routes are similar to both the proposed algorithm and Dijkstra's algorithm for source – destination pair. Dijkstra's algorithm is always known to offer the shortest path [9]. Fig. 17(a) shows the route returned by the proposed algorithm that gives the optimal value of end-to-end delay.

5.2 Simulation results for Network with 100 nodes

Fig. 18 shows the varied curves of end-to-end delay, and energy consumption with the increasing generations. Simulations are performed with a network size of 100 nodes. In our work, population size *pop* and network size are taken as the same. The Fig. 18 indicates that the end-to-end delay is 1.7359 μs, when generation=1. For generation=2 and 3, end-to-end delay is 1.6567 μs, for generation=4 and onwards, end-to-end delay becomes 1.6504 μs. At this stage (generation=4), the solution converges to an optimal value. Similar behavior can be observed in the case of energy consumption. So, it can be concluded that the proposed routing algorithm makes effort to maximize the probabilities of meeting energy consumption, and end-to-end delay within a few generations.

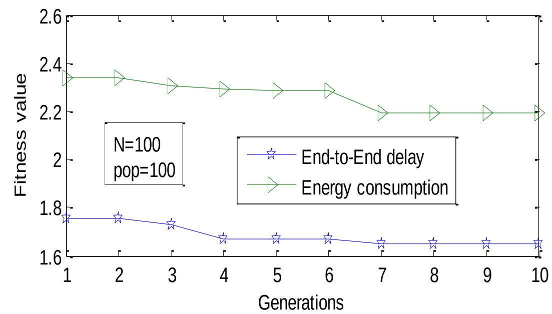


Fig.18. Convergence of the solution for different fitness functions.

Fig. 19(a)-(f) show the variation in the fitness values (end-to-end delay) of individuals in the population. Fig. 19(a) depicts the up and down values of the fitness function, and the maximum fitness value is about to 3.4. This is because, at the first generation, the initial population of size 100 is chosen randomly where all possible paths in FZ are taken which also includes the path with maximum end-to-end delay. From Fig. 19(b), it can be seen clearly that for generation=2, from starting to half of the population, the difference between fitness values are decreasing, and the maximum fitness value become to 2.4. This is because; our algorithm adopts a truncation selection method that selects only 50% high quality of individuals in the population for

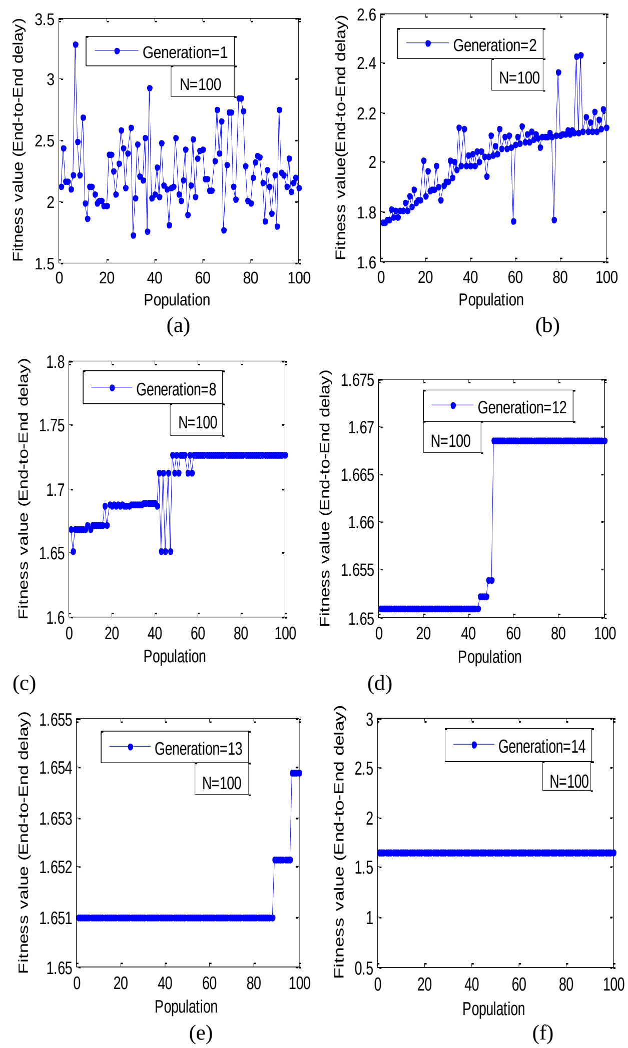


Fig.19.Results for End-to-End delay when number of generations varies. (a) Generation=1. (b) Generation=2. (c) Generation=8. (d) Generation=12. (e) Generation=13. (f) Generation=14.

crossover to produce a new population. Accordingly, in Fig. 19(c) - (e), it is observed that at each next generation, the fitness value is shrinking towards an optimal solution. In Fig. 19(e), it can be seen that at generation=14, all the individuals producing the same value that is the global solution.

5.3 Comparison of the fitness values for various algorithms for network size of 500

Fig. 20 compares the fitness values of end-to-end delay, computed by the algorithms. The fitness values returned by the proposed algorithm, Ahn-Ramakrishna's algorithm, Dijkstra's algorithm, DIR, and MFR (cf., Table IV) are represented in Fig. 20. In the proposed algorithm, and Ahn-Ramakrishna's algorithm, the fitness values of end-to-end delay decreases with the generations. In our case, the fitness value decreases up to generation 4, and after generation 4, it remains consistent. It means the optimal solution converges at generation 4. For Ahn-Ramakrishna's algorithm, the optimal solution converges at generation 6. When network size is 100, for example,

TABLE IV

EXPERIMENTAL RESULTS OF FITNESS FUNCTIONS OF COMPARING ALGORITHMS IN DIFFERENT NETWORK SIZE, WHEN N=50 , N=100 AND N=500.

Algorithms	End-to-End delay (µs)			Energy consumption (mJ)		
	N=50	N=100	N=500	N=50	N=100	N=500
Proposed Algorithm	1.7060	1.6504	1.3805	2.3373	2.1465	1.9212
Ahn-Ramakrishna's algorithm	1.7060	1.6504	1.3805	2.3373	2.1465	1.9212
Dijkstra's Algorithm	1.7060	1.6504	1.3805	2.3373	2.1465	1.9212
DIR	1.8049	1.6601	1.4505	2.6226	2.3835	2.1425
MFR	2.1632	1.7697	1.5128	3.3496	2.6165	2.5125
RRDLA	1.7360	1.6600	1.4305	2.5325	2.2475	2.0512

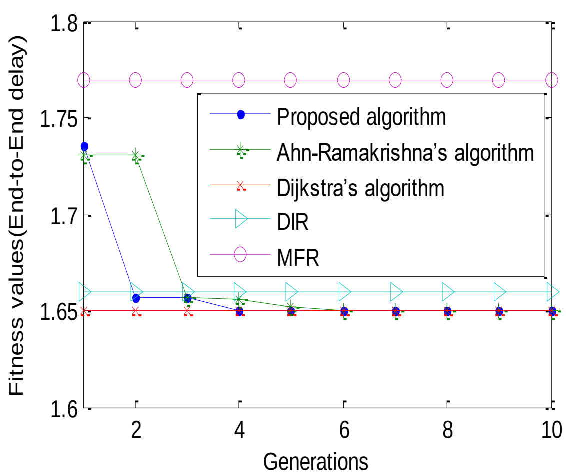


Fig.20. Convergence property and variance in the fitness values of ach algorithm.

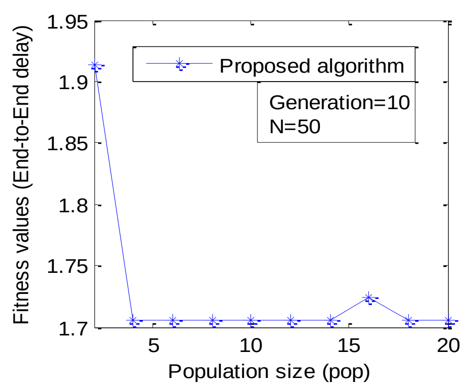


Fig.21. End-to-End delay when varying the population size.

the optimal value of end-to-end delay for the proposed algorithm Ahn-Ramakrishna's algorithm and Dijkstra's algorithm are 1.6504 µs. From Fig. 20, it is observed that the proposed algorithm converges to a solution that is exactly the same as returned by Dijkstra's algorithm that is known always to offer the optimal solution [9]. Therefore, the proposed algorithm exhibits better performance as compared to other algorithms. Also, it is noted that the optimal value of the objective function is obtained at a small number of generations which shows the fast rate of convergence. In our work, the convergence speed is improved by 33% as compared to Ahn-Ramakrishna's algorithm.

5.4 The impact of population size on the quality of solution

In this section, the impact of population size is investigated with network size of 50. To find the appropriate value of population size, the fitness function is evaluated for generations 1 to 10, and the best fitness value is picked among the generations. The simulation experiments are performed for different values of population size and obtained results are compared with the desired solution, so that, suitable population size can be found. From the Fig. 21, it can be seen that when the network size is 50, the best fitness value is obtained for population size ranging from 4 to 14. It must be noted that as the population size increases to 16, the best fitness value is not obtained. Therefore, it can be concluded that we cannot always obtain the best solution even for a large value of population size. Finally, through repeated experiments, suitable population size can be discovered.

TABLE VI

COMPARISON BETWEEN EXPERIMENTAL RESULTS AND ANALYTICAL RESULTS FOR VARIOUS PARAMETER OF PROPOSED ALGORITHM, WHEN N=50,100,500.

QoSParameters	Experimental results N=50	Analytical Results N=50	Experimental Results N=100	Analytical Results N=100	Experimental Results N=500	Analytical Results N=500
End-to-End delay (μs)	1.7060	1.3335	1.6504	1.1112	1.3805	0.8125
Energy Consumption (mJ)	2.3373	1.5361	2.1465	1.2672	1.9212	1.1500

The results obtained through simulation, for different routing algorithms are summarized in Table IV. From this table, when network size is 50, it can be seen that the optimal values for QoS parameters, i.e., end-to-end delay, and energy consumption are 1.7060, and 2.3373, respectively. In the case, when the network size is 100, the optimal fitness values for the indicated parameters are 1.6504, and 2.1465, respectively. In our case, the values of the indicated parameters are smaller than that of DIR, and MFR for both 50 and 100 sizes of the network. As expected, the proposed algorithm outperforms the well-known algorithms considered for comparison.

In order to compare the optimal route and path length of the proposed algorithm with that of Ahn-Ramakrishna's algorithm, Dijkstra's algorithm, DIR, and MFR, the results obtained through simulations are presented in Table IV. The results present here, are only for the network size of 50 and 100. From this Table, it can be seen that both the proposed algorithm and Ahn-Ramakrishna's algorithm found the same optimal route for the end-to-end delay, and energy consumption. This is because; both the algorithms used a genetic approach to their algorithm. Further, it is also noted that the desired solutions are not affected much to path length. When N=50, for example, the paths obtained by our algorithm and DIR are 8 15 29 16 46 40 23 13 and 8 39 29 16 46 40 23 13, respectively. We can see that the path length for both the algorithms is same, i.e., 7, but the fitness values obtained by these algorithms are not same. Only our algorithm gives the better results. (cf., Table V)

Table VI presents the experimental and analytical results of the proposed algorithm for various metrics such as end-to-end delay, and energy consumption, when network sizes are 50 and 100. The analytical results of end-to-end delay and energy consumption are obtained by (17) and (21), respectively.

TABLE V
EXPERIMENTAL RESULTS OF SHORTEST PATHS, AND PATH LENGTH OF COMPARING ALGORITHMS FOR DIFFERENT NETWORK SIZE, WHEN N=50 AND N=100.

Algorithms	Optimal route N=50	Path length N=50	Optimal route N=100	Path length N=100
Proposed Algorithm Ahn-Ramakrishna 'salgorithm Dijkstra's Algorithm	End-to-End delay 8 15 29 16 46 40 23 13	7	End-to-End delay 8 25 97 78 83 76 37 13	7
	Energy Consumption 8 15 26 29 16 46 40 23 13	8	Energy Consumption 8 25 97 69 57 48 61 60 76 14 37 13	11
DIR	8 39 29 16 46 40 23 13	7	8 25 97 78 35 28 43 89 13	8
MFR	8 39 29 10 11 14 6 9 13	8	8 25 97 78 28 14 89 13	7

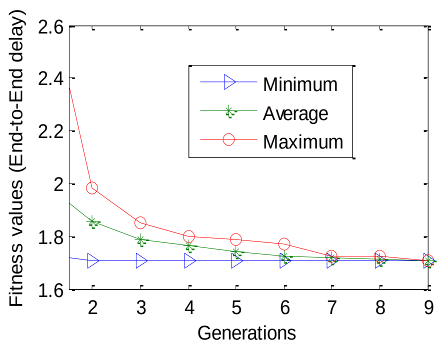


Fig.22.Comparison of maximum, average, and minimum fitness values

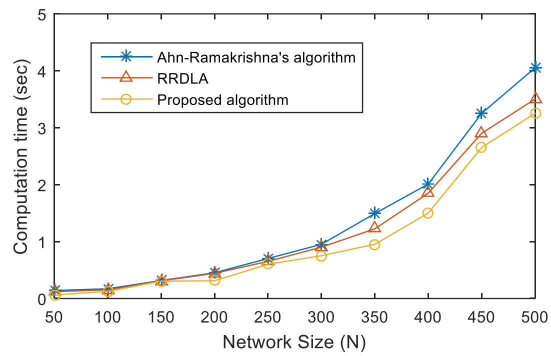


Fig. 24. Comparison of computation time

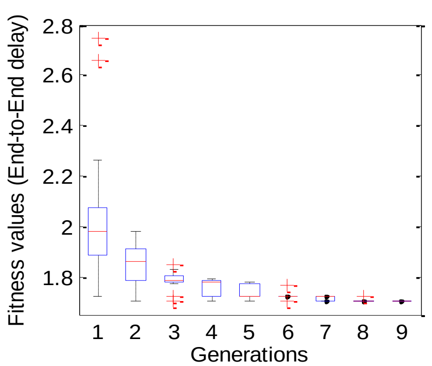


Fig.23. Box plot of the fitness values for End-to-End delay

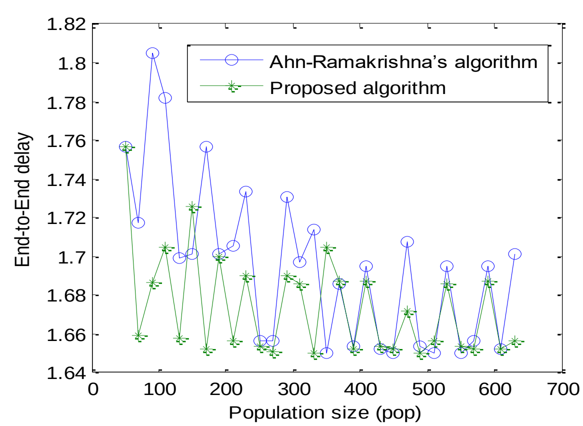


Fig. 25.Comparison of end-to-end delay with varying population size.

In Table VI, the experimental results for the end-to-end delay is presented. These results are obtained from generation 1 to 9 when the network size is 50. Here, the population size is set to the same as network size. The best solution, which is 1.7060, is obtained at generation 2. At generation 9, all the individuals generate the same fitness values. It is observed that the average fitness value of the individuals decreases at each next generation. Because in our algorithm, we used truncation selection where at each generation, the best 50% individuals are picked to generate new generation.

Fig. 22 depicts the comparison of the minimum, average, and maximum fitness values of end-to-end delay when the network size is 50. It is noted that at the first generation, the minimum, average and highest fitness values are 1.725, 1.995 and 2.748, respectively, while at generation 9, the minimum, average and highest fitness values are same, i.e., 1.706. From the Fig.22, it can be

TABLE VII

FITNESS VALUES OF END-TO-END DELAY OF ALL CHROMOSOMES (INDIVIDUALS) FOR GENERATION 1 TO 9, WHEN NETWORK SIZE N=50.

Chromosomes	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6	Gen 7	Gen 8	Gen 9
1	2.2633	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060	1.7060	1.7060
2	1.9136	1.7246	1.7060	1.7060	1.7060	1.7060	1.7060	1.7060	1.7060
3	1.9985	1.7246	1.7246	1.7060	1.7246	1.7060	1.7060	1.7060	1.7060
4	2.1875	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060	1.7060	1.7060
5	1.8297	1.7843	1.7776	1.7716	1.7246	1.7060	1.7060	1.7060	1.7060
6	2.0557	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060	1.7060
7	1.9853	1.7873	1.7246	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
8	1.9853	1.7847	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060	1.7060
9	1.7246	1.7830	1.7246	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
10	1.9377	1.7856	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060	1.7060
11	1.9853	1.7873	1.7830	1.7776	1.7060	1.7246	1.7060	1.7060	1.7060
12	1.9853	1.7873	1.7830	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
13	1.7923	1.7923	1.7843	1.7776	1.7060	1.7246	1.7060	1.7060	1.7060
14	1.9157	1.7923	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
15	1.7847	1.7976	1.7869	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
16	1.7873	1.8002	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
17	1.9872	1.8049	1.7847	1.7246	1.7246	1.7246	1.7246	1.7060	1.7060
18	1.9898	1.8260	1.7847	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
19	2.1789	1.8260	1.7856	1.7246	1.7246	1.7246	1.7246	1.7060	1.7060
20	1.9286	1.8260	1.7856	1.7246	1.7246	1.7246	1.7060	1.7060	1.7060
21	1.7856	1.7060	1.7873	1.7776	1.7246	1.7060	1.7246	1.7060	1.7060
22	1.9720	1.8297	1.7873	1.7776	1.7246	1.7246	1.7246	1.7060	1.7060
23	1.9128	1.8895	1.7873	1.7976	1.7246	1.7060	1.7246	1.7060	1.7060
24	1.9128	1.8789	1.7873	1.7791	1.7246	1.7246	1.7246	1.7060	1.7060
25	1.9975	1.8895	1.7873	1.7830	1.7246	1.7246	1.7246	1.7060	1.7060
26	1.8895	1.8895	1.7873	1.7830	1.7246	1.7246	1.7246	1.7060	1.7060
27	1.9786	1.9000	1.7927	1.7830	1.7246	1.7246	1.7246	1.7060	1.7060
28	1.9817	1.8974	1.7923	1.7830	1.7246	1.7246	1.7246	1.7060	1.7060
29	2.0523	1.9074	1.7927	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060
30	2.0523	1.9128	1.7923	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060
31	2.6612	1.9101	1.7953	1.7869	1.7246	1.7246	1.7246	1.7060	1.7060
32	2.7480	1.9128	1.7923	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060
33	2.1420	1.9322	1.7791	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060
34	2.1420	1.9136	1.7976	1.7843	1.7246	1.7246	1.7246	1.7060	1.7060
35	1.8974	1.9157	1.8002	1.7847	1.7716	1.7716	1.7246	1.7060	1.7060
36	1.8789	1.9157	1.8002	1.7847	1.7716	1.7246	1.7246	1.7060	1.7060
37	2.1113	1.8049	1.8260	1.7847	1.7776	1.7246	1.7246	1.7246	1.7060
38	1.9667	1.9286	1.8049	1.7847	1.7776	1.7246	1.7246	1.7060	1.7060
39	1.9348	1.9237	1.8260	1.7856	1.7776	1.7246	1.7246	1.7246	1.7060
40	2.1541	1.9348	1.8049	1.7856	1.7776	1.7246	1.7246	1.7060	1.7060
41	1.8260	1.8506	1.8049	1.7856	1.7776	1.7246	1.7246	1.7246	1.7060
42	1.8260	1.9377	1.8260	1.7856	1.7776	1.7246	1.7246	1.7246	1.7060
43	2.0307	1.9667	1.8049	1.7869	1.7776	1.7716	1.7246	1.7246	1.7060
44	1.7843	1.9667	1.8260	1.7869	1.7776	1.7246	1.7246	1.7246	1.7060
45	1.7246	1.9720	1.8234	1.7873	1.7843	1.7716	1.7246	1.7246	1.7060
46	2.0769	1.9720	1.8260	1.7873	1.7791	1.7246	1.7246	1.7246	1.7060
47	2.0971	1.7923	1.7246	1.7873	1.7830	1.7246	1.7060	1.7246	1.7060
48	2.0894	1.9786	1.8297	1.7873	1.7830	1.7246	1.7246	1.7246	1.7060
49	2.1073	1.9157	1.8297	1.7873	1.7776	1.7246	1.7060	1.7246	1.7060
50	1.8002	1.9817	1.8506	1.7873	1.7830	1.7246	1.7246	1.7246	1.7060

Fig. 23 represents the results given in Table VII, in the form of the box plot. It depicts graphically the data distribution through quartiles in a convenient way. The red line inside the box represents the median value of the data. From the Fig.23, it can be seen that the median value decreases as the number of generations increase. It is observed that at generation=1, 25% - 75% of fitness values lie between 1.9 and 2.1. Large size of the box represents the more variation in the results. As the number of generations increases, the size of the box decreases which shows the convergence of the optimal solution. It is also noted that at generation=9, there is only one line representing the global solution.

5.5 The comparison of computation time

In order to compare the performance of the proposed algorithm with that of Ahn-Ramakrishna's algorithm, and RRDLA considering computation time as a metrics, the result is presented in Fig. 24. The result is the consideration of scaled network size. We mean here, the simulations are conducted for the network size of 50 to 500 nodes. It can be observed that the computation time of the proposed algorithm is lesser as compared to state of the art techniques particularly visible for larger network size. For example, for network size N=400, the computation time for the proposed GA is 1.5 seconds, whereas for RRDLA, and Ahn-Ramakrishna's algorithm are 1.8 and 1.9 respectively which is a performance benefits of approximately 20% and 26% respectively. It is because of the reason that the proposed GA has smaller search space for finding the solution. Thus, proposed GA shows better performance as compared to the considered state of the art techniques.

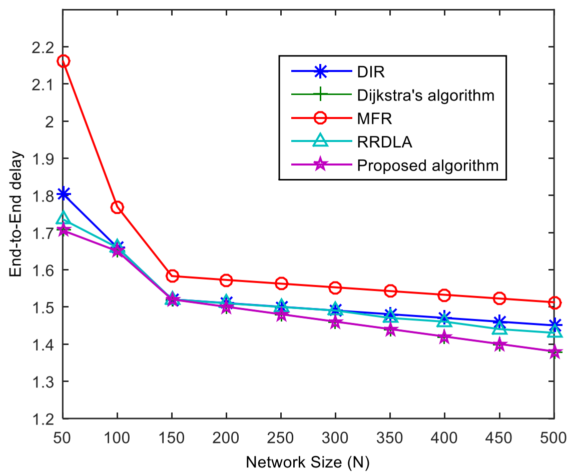


Fig. 26.Comparison of end-to-end delay (μs) with varying network size.

5.6 End-to-End delay comparison with varying population size, and network size

The results in respect of performance comparison for end-to-end delay metrics are shown in Fig. 25 and 26. It can be observed that the obtained average end-to-end delay for the proposed GA is lower than that of Ahn-Ramakrishna's algorithm (Fig. 25). Here, to analyze the performance, population size is varied and number of generation is fixed. In order to compare the performance of the proposed GA with that of DIR, MFR, Dijkstra's algorithm, and RRDLA considering end-to-end delay, the obtained experimental result is presented in Fig. 26. The results are obtained for scaled network size of up to 500 nodes. Here, for a network size $N = 100$, average end-to-end delay for the proposed GA, RRDLA, DIR, and MFR are 1.674, 1.685, 1.695 1.784μs, which is a performance gain or reduction in delay of 5%, 11%, and 29%, respectively. The end-to-end delay reduces to 1.375, 1.445, 1.465 , and 1.524 μs for scaled network size $N = 500$, which is again a performance gain or reduction in delay of 5%, 6%, and 11%, respectively. Thus, reduction in end-to-end delay of the proposed GA is quite visible as performance gain in comparison with those of state of the art techniques.

5.7 Analysis of proposed GA through ANOVA test

In this section, the performance of the proposed GA is compared with that of Ahn-Ramakrishna's algorithm, and MFR using ANOVA (Analysis of Variance). Here, ANOVA test is performed for varying network size N=50 to 150 into three samples. In Fig. 27, box-1, box-2, and box-3, represent the results of proposed GA, Ahn-Ramakrishna's algorithm, and MFR, respectively. Fig.28 represents the ANOVA table. From the Fig. 27, it can be seen that the median value for the proposed GA is less, and most of the values lie near to median value, as compared to Ahn-Ramakrishna's algorithm, and MFR. It is justified that that proposed GA gives better results as compared to Ahn-Ramakrishna's algorithm.

From the ANOVA table there is difference between mean square(MS) between group-variations (Columns) and mean square within-groups (Error), also the F-Statistic ratio (MS(columns)/MS(Error)) is greater than one, so it rejects the Null hypothesis which means there is difference between mean (μ_0) of samples taken. As Calculated probability is much less the

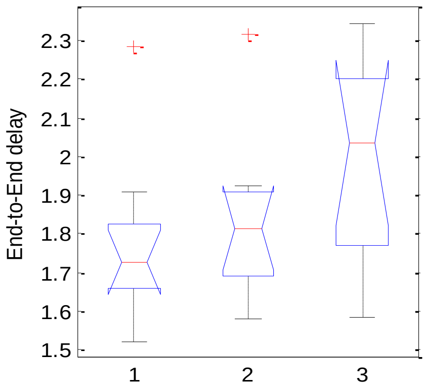


Fig. 27.ANOVA graph

Source	SS	df	MS	F	Prob>F
Columns	0.28075	2	0.14037	2.76	0.0109
Error	1.37117	27	0.05078		
Total	1.65192	29			

Fig. 28.ANOVA Table

general significance value (α=0.05) ie.0.0109<0.05 which also gives the strong evidence against the Null hypothesis.

6. Conclusion

This paper presented a GA based routing to determine the optimal route in a set of multiple routes with minimum end-to- end delay, and energy consumption requirements. In order to reduce the computation cost, a sub-network known as the forward zone is constructed. The algorithm searches the optimal solution in a small set of the search space. The infeasible chromosomes have also been prevented at the stage: chromosomes formation. Further, a specialized genetic algorithm with various fitness functions is designed. Extensive simulations are conducted in this paper to evaluate the performance of the proposed GA. Also, to validate the experimental results with the analytical results, various mathematical models are developed. Simulation studies show that introducing the concept of forward zone enhances the performance of GA in respect of computation cost. The convergence speed of proposed algorithm is better than Ahn-Ramakrishna's algorithm and RRDLA. In our work, the convergence speed and average computation time are improved by 33% and 22% respectively, as compared to Ahn-Ramakrishna's. It is seen that the optimal solution obtained by both proposed algorithm and Ahn-Ramakrishna's algorithm is the same, but the proposed algorithm takes less number of generations as compared to later one. Simulation experiments indicated that our algorithm gives better solutions than that of DIR, MFR algorithm, and RRDLA for different network size. In future work, GA based approaches will be investigated for the optimal size of the forward zone without compromising the quality of solutions. In addition, other evolutionary approach such as PSO can be considered to solve the indicated problems.

References

[1] I.F. Akyildiz, W. Su, Y.Sankarasubramaniam, and E. Cayirci,"A survey on sensor networks," *IEEE Communications magazine*, vol.40, no. 8,pp.102-114, Aug. 2002.

[2] Aanchal, Kumar, S., Kaiwartya, O. et al., “Green computing for wireless sensor networks: Optimization and Huffman coding approach“, Peer-to-Peer Netw. Appl, vol. 10, issue 3, pp 592–609, 2017
[3] Kumar S, Lobiyal D. L, “Sensing Coverage Prediction for Wireless Sensor Networks in Shadowed and Multipath Environment,” The Scientific World Journal, vol. 2013, Article ID 565419, 11 pages, Sept- 2013
[4] Kumar S, Lobiyal D. L,, "Impact of Interference on Coverage in Wireless Sensor Networks", Wireless Personal Communications, Springer, Jan- 2014, vol. 74, Issue 2, pp. 683-701
[5] Khatri, A.; Kumar, S.; Kaiwartya, O.; Aslam, N.; Meena, N., Abdullah, A.H., “Towards green computing in wireless sensor networks: Controlled mobility–aided balanced tree approach”, Int. J. Commun. Syst. 2017. https://doi.org/10.1002/dac.3463
[6] Omprakash Kaiwartya, Sushil Kumar, Abdul Hanan Abdullah, “Analytical model of Deployment Methods for Application of Sensors in non-Hostile Environment”, Wireless Personal Communications, Springer, November 2017, Volume 97, Issue 1,
[7] A. P. Bhondekar, R. Vig, M. L. Singla, C. Ghanshyam, and P. Kapur, “Genetic algorithm based node placement methodology for wireless sensor networks,” in *Proc of the International Multiconference of Engineers and Computer Scientists*, vol. 1, 2009, pp. 18-20.
[8] Kumar, V.; Kumar, S. Energy balanced position-based routing for lifetime maximization of wireless sensor networks. Ad Hoc Netw. 2016, 52, 117–129.
[9] C.W. Ahn, and R. S. Ramakrishna, “A genetic algorithm for shortest path routing problem and the sizing of populations,” *IEEE Trans. Evol.Comput*., vol. 6, no. 6, pp. 566-579, Dec. 2002.
[10] S. A. Madani, D. Weber, and S. Mahlknecht, “TPR: Dead end aware table less position based routing scheme for low power data-centric wireless sensor networks,” in *Industrial Embedded Systems, SIES 2008, Int. Symposium* , pp. 149-154, Jun. 2008.
[11] E. Zitzler, Evolutionary algorithms for multiobjective optimization: Methods and applications, vol. 63, Ithaca: Shaker, 1999.
[12] J. Inagaki, M. Haseyama, and H. Kitajima, “A genetic algorithm for determining multiple routes and its applications,” in *Proc. IEEE Int. Symp. Circuits and Systems*, 1999, pp. 137–140.
[13] S. Yang, H. Cheng, and F. Wang, “Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks,” *IEEE Trans. On Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol.40, no.1, pp.52-63, Jan.2010.
[14] H. Saleet, R. Langar, K. Naik, R. Boutaba, A. Nayak, and N. Goel, “Intersection-based geographical routing protocol for VANETs: a proposal and analysis,” *IEEE Trans. on Vehicular Technology*, vol. 60, no. 9, pp. 4560-4574, Nov.2011.
[15] T. Blickle, Theory of evolutionary algorithms and application to system synthesis, vdfHochschulverlag AG, 1997.
[16] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerischemathematik*, vol. 1, no. 1, pp. 269-271, 1959.
[17] M. Barbehenn, “A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices,” *IEEE transactions on computers*, vol. 47, no.2, pp. 263, Feb.1998.
[18] E. Kranakis, H. Singh, and J. Urrutia, “Compass routing on geometric networks,” in *Proc. 11 th Canadian Conference on Computational Geometry*, 1999, pp.1-4.
[19] I. Stojmenovic and X. Lin, “Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 12, no.10, pp. 1023-1032, Oct. 2001.
[20] I. Stojmenovic, A. P. Ruhil, and D. K. Lobiyal, “Voronoi diagram and convex hull based geocasting and routing in wireless networks,” *Wireless communications and mobile computing*, vol. 6, no. 2, pp. 247-258, Feb. 2006.
[21] H. Takagi and L. Kleinrock, “Optimal transmission ranges for randomly distributed packet radio terminals,” *IEEE Trans. on Communications*, vol. 32, no. 3, pp. 246-257, Mar. 1984.
[22] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: A brief survey,” *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 2, pp. 262-267, Mar. 2011.
[23] M. K. M. Ali and F. Kamoun, “Neural networks for shortest path computation and routing in computer networks,” *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 941-954, Nov. 1993.

[24] W. Stalling, High-Speed Networks: TCP/IP and ATM Design Principles. Englewood Cliffs, NJ: Prentice-Hall, 1998.
[25] N. Shimamoto, A. Hiramatsu, and K. Yamasaki, “A dynamic routing control based on a genetic algorithm,” in *Proc. IEEE Int. Conf. Neural Networks*, 1993, pp. 1123–1128.
[26] Z. Xiawei, C. Changjia, and Z. Gang, “A genetic algorithm for multicasting routing problem,” in *Proc. Int. Conf. Communication Technology (WCC-ICCT 2000)*, 2000, pp. 1248–1253.
[27] M. Munemoto, Y. Takai, and Y. Sato, “A migration scheme for the genetic adaptive routing algorithm,” in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, 1998, pp. 2774–2779.
[28] T. C. Hou and V. O. Li, “Transmission range control in multihop packet radio networks,” *IEEE Trans. on Communications*, vol. 34, no. 1, pp. 38-44, Jan. 1986.
[29] J. N. Al-Karaki and A. E. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless communications*, vol. 11, no. 6, pp. 6-28, Dec. 2004.
[30] N. Beijar, “Zone routing protocol (ZRP),” Networking Laboratory, Helsinki University of Technology, Finland, pp.1-12, Apr. 2002.
[31] H. Mostafaei, “Energy-efficient algorithm for reliable routing of wireless sensor networks,” *IEEE Transactions on Industrial Electronics*, vol. 66, no.7, pp. 1–10, Sept. 2018.
[32] X. Zhang, C. Wang, L. Tao, “An opportunistic packet forwarding for energy-harvesting wireless sensor networks with dynamic and heterogeneous duty cycle”, *IEEE Sensors Lett.*, vol. 2, no. 3, Sep. 2018.
[33] X. Lai, X. Ji, X. Zhou, L. Chen, “Energy efficient link-delay aware routing in wireless sensor networks”, *IEEE Sensors J.*, vol. 18, no. 2, pp. 837-848, Jan. 2018.
[34] F. A. Khan, M.Khan, M.Asif, A.Khalid, I.U. Haq, “Hybrid and Multi-Hop Advanced Zonal-Stable Election Protocol for Wireless Sensor Networks”, *IEEE Access*, vol.7, pp. 25334-25346, Feb. 2019.
[35] L.Lin and M.Gen, “Node-based genetic algorithm for communication spanning tree problem,” *IEICE transactions on communications*, vol. 89, no. 4, pp.1091-1098, Apr.2006.
[36] L. Lin and M.Gen, “Priority-based genetic algorithm for shortest path routing problem in OSPF,”In *Intelligent and Evolutionary Systems*, Springer Berlin Heidelberg, , pp. 91-103, 2009.
[37] L.Lin and M.Gen, “Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation,” *Soft Computing*, vol.13, no. 2, pp.157-168, Jan.2009.
[38] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proc. IEEE 33rd annual Hawaii Int. Conf. on System sciences*, Jan. 2000, pp. 1-10.
[39] http://en.wikipedia.org/wiki/Path_loss
[40] M. Heissenbüttel and T. Braun, “A novel position-based and beacon-less routing algorithm for mobile ad-hoc networks,” in *ASWN*, Vol. 3, pp. 197-210, Jul. 2003.