JWSN 2016, 3, 1-0014

Journal of Wireless Sensor Networks

ISSN: 2001-6417

www.wsn-journal.com

Article

An Energy Efficient Evo-Fuzzy Sleep Scheduling Protocol for Stationary Target Coverage in Wireless Sensor Networks

Srinjoy Ganguly

Jadavpur University – Dept. of Electronics & Telecommunication Engineering
srinjoy_ganguly92@hotmail.com

Received: 08th August 2015 / Accepted: 11th December 2015/ Published: 21st January 2016

Abstract: Target coverage is a fundamental problem that needs to be addressed in sensor networks for a variety of applications such as environment monitoring and surveillance purposes. A typical approach to prolong network lifetime would entail the partitioning of the sensors capable of sensing the targets, in a network for target monitoring into several disjoint subsets such that each subset can cover all the targets. Thus, each time only the sensors in one of such subsets are activated. In this paper, we have proposed a novel sleep scheduling protocol, abbreviated as EEFSSP, based on this concept which incorporates three novel features. Firstly, it paves way for an equitable distribution of nodes while forming cover sets through the proposed CSGH heuristic. Secondly, it schedules the cover sets using an evolutionary approach with the objective being to optimize the maximum breach interval. Thirdly, the EEFSSP introduces a novel routing protocol abbreviated as DFPRP to establish routes to transfer data packets to the Base Station, with the objective being to ensure energy-efficiency and minimize the number of packet drops. We finally conduct experiments by simulation to evaluate the performance of the proposed scheme under various conditions, and compare its performance with other relevant protocols. The experimental results show that the proposed scheme clearly outperforms its peers by delivering a much longer network lifetime and minimizing the number of packet drops.

Keywords: Wireless Sensor Networks, differential evolution, fuzzy rule base, evolutionary algorithms, data packet routing, target coverage, EEFSSP, DFPRP, CSGH, HDDE

I. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] are mobile networks which comprise of a large number of autonomous sensor nodes deployed into a large geographical area. Recent advances in MEMS design have made it possible to construct compact and inexpensive wireless sensors. Each sensor has wireless communication capability and some level of intelligence for signal processing and networking of the data. Networks consisting of sensors have received significant attention due to their potential applications from civil to military domains. In a typical WSN application, the source nodes collect data from a phenomenon and disseminate them toward the sink node using multi-hop communication. The sensor nodes are typically equipped by power-constrained batteries, which are often difficult and expensive to be replaced once the nodes are deployed. Therefore, it is a critical consideration on reducing the power consumption in the network design.

In this paper, we consider a sensor network used for monitoring stationary targets. The network comprises of a number of sensor nodes, deployed into the field containing the targets, and a Base Station (BS) for centralized control. Each node can exist in two state, namely, the low-energy consumption sleep state [2] and the active state [3], where the node actively participates in the network in the latter case while the reverse holds true in the former. Also, each node has a definite one-hop communication range and a sensing range, where the former may not be equal to the latter. Hence, a set of nodes is responsible for sensing any event which occurs at the targets while another set is responsible for forwarding the data from these nodes to the BS. One efficient method of reducing the energy consumption of sensors and thereby prolonging the network lifetime is to partition the sensors in the network into multiple subsets (sensor covers) such that each subset can cover all the targets or as many targets as possible, in case there is a limit on the number of sensors in each cover set or no cover set covering all the targets is obtainable (due to node death or other related issues). Thus, each time only one sensor cover is activated for a certain period and only the sensors in the active sensor cover are in active mode, while all the other sensors are in sleep mode to save energy. Hence in order to ensure optimal coverage of the targets, an optimal sleep schedule for the set of covers ought to be obtained such that the longest time any target is uncovered, i.e. the breach interval is minimized. This is referred to as the Cover Set Scheduling Problem [4]. Therefore, obtaining the optimal schedule of cover sets for efficient target coverage is tantamount to solving the Cover Set Scheduling Problem. Each cover set in a given schedule will occupy a slot and a new schedule is computed for every new round of data gathering (from the targets).

Another important problem which needs to be taken care of is the development of an energy-efficient, secure protocol to route data packets from the nodes to the BS. Energy efficiency is one of the most critical issues in WSNs. With the current available technology, sensors are battery powered and have a limited weight. These characteristics globally affect the application lifetime. As the WSN remains operational, due to energy consumption for sending and relaying the environmental data, sensor nodes gradually run short of battery and ultimately they become un-operational. Energy-efficient, secure routing of data packets can be achieved by establishing routes which take various factors such as congestion, distance and node energy into consideration. Unless energy efficiency and security is maintained in the routing procedure, the network will be subject to various performance bottlenecks such as buffer overflow, increased latency, packet drop, wastage of energy, deterioration of QoS and network throughput. Hence, the afore-mentioned objectives could be achieved by virtue of a distributed protocol which would assimilate the most favorable links so as to obtain an optimal route to forward data packets to the BS.

To integrate this step with the rest of the protocol, the nodes present in the route leading from a node monitoring a target (sensing node) to the BS may enter the active state only when the corresponding sensing node is active, while the rest remain in the sleep state.

In this paper we have proposed a new sleep scheduling scheme for efficient coverage of a set of targets in a sensor network. The key components proposed scheme and the corresponding novel features incorporated in each are: (i) A novel heuristic to generate cover sets at the beginning of each round. (ii) An evolutionary algorithm to efficiently schedule the covers, i.e. which cover will operate in which slot. (iii) A fuzzy node probability based protocol to find out an optimal path to route the data packets to the BS. (iv) A sleep scheduling scheme to ordain which subset of nodes is in the active state in a slot, with the others in the sleep state, based upon the node marked out in steps II and III. Hence, we propose an Energy-efficient Evo-Fuzzy Sleep Scheduling Protocol (EEFSSP) for efficient monitoring of stationary targets in WSNs. To verify the efficacy of the protocol, the performance of the overall algorithm and its individual components is compared with other relevant protocols on a wide array of test cases. The rest of the paper has been organized as follows. In section II, we present a brief discussion of the related works. The necessary models and formulations are explained in section III. The proposed algorithm is detailed in section IV. The experimental results and related discussions are presented in section V and we finally conclude the paper in section VI.

II. RELATED RESEARCH WORKS

Sensor nodes have size, weight and cost restrictions, which have a direct impact on resource availability. In recent years, tremendous effort on network lifetime maximization and energy efficiency has been taken for target coverage, sensor connectivity and network fault tolerance [5 - 8]. As replacing the battery is not feasible low power consumption is one of the chief aims while designing protocols for a sensor network [9]. Hence several power-efficient schemes have also been proposed in literature [10].The LEACH protocol [11] presents an elegant solution to this energy utilization problem where nodes are randomly selected to collaborate to form small number of clusters and the cluster heads take turn in transmitting to the base station during a data gathering cycle. It improves energy cost per round by a factor of 4 for a 100 node network as compared to a direct approach where individual nodes transmit directly to the base station. The PEGASIS protocol [12] is a further improvement upon the LEACH protocol where a chain of nodes is formed which take rounds in transmitting data to the BS. A multi path routing protocol based on dynamic clustering and ACO, is described in MRP [13], which improves the efficiency of data aggregation, thereby reducing the energy consumption. Several other relevant data gathering and/or routing protocols have been discussed in [14 - 20]. Without enacting any energy saving technique during WSN operations, the radio transceiver would typically consume more energy than any other hardware component onboard a sensor node. The study in [21] examined the pattern of energy expenditure in wireless communications for a practical WSN and reported that the bulk of the energy is consumed during idle listening, while the dominant traffic type processed by individual nodes are overheard packets. Sleep scheduling can be used to address the afore-mentioned problem. While the main implication of sleep scheduling at the MAC layer is the shortening of the time the radio transceiver is engaged in idle listening, incidences of overhearing can also be reduced as sleeping nodes are no longer eavesdropping on the wireless medium. Several such sleep scheduling schemes designed to enhance network lifetime in WSNs have been discussed in [22 - 26]. Ideally a cross-layer approach should be adopted while deciding upon sleep schedules because this ensures that the various sub-layers can be managed simultaneously.

In recent years, tremendous effort on network lifetime maximization and energy efficiency has been taken for target coverage, sensor connectivity and network fault toleration. Cardei et. Al [27] dealt with the target coverage problem by organizing sensors into disjoint sensor covers, and allowing only one of the covers to be activated at any given instant of time. This led to a significant improvement in the network lifetime. Chakrabarty et al [28] employed grid coverage and presented a linear integer programming solution for the target coverage problem. Dong et al [29] considered load balancing while guaranteeing that each target is monitored by at least one sensor with the objective being to minimize the maximum energy consumption of sensors for, and proposed centralized and distributed algorithms for the same. Wang et al [30] adopted two distinct overage methods to explore the sensor density required for guaranteeing a localization error bound over the field. Cai et al [31] studied the problem of multiple directional sensor covers by proving its NP-completeness, and proposed heuristic solutions for the same. Lu et al [32] addressed the problem by adjusting the sensing ranges of sensors and devised a distributed algorithm to determine the sensor range of each sensor. Li et al [33] considered the k-connected target coverage problem by showing its NP-hardness, and they instead designed two heuristic algorithms for the problem. Zhou et al [34] introduced the k_1-connected, k_2-cover problem and proposed a distributed algorithm for the same. The cover set scheduling problem arising in WSNs has been discussed and a proof for its NP-hardness has been stated by Rossi et al in [35]. Liu et al [36] proposed a Steiner-tree based approach for the coverage problem in sensor networks by taking both residual energy and coverage ability of the sensors into account. Several other algorithms for the optimal scheduling of cover sets in directional/non-directional sensor networks have been discussed in [37 - 39].

III. RELATED MODELS & FORMULATIONS

A. Network Model

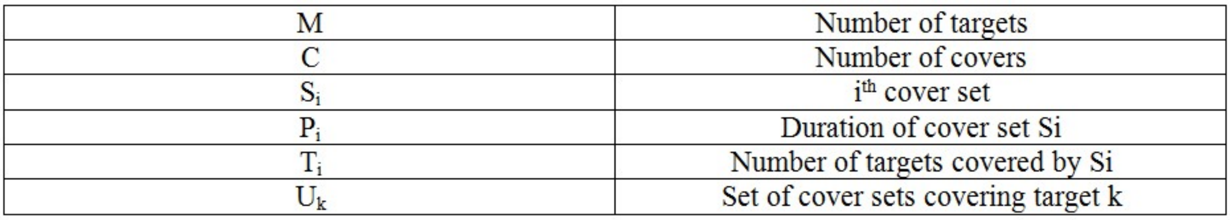
In this paper, we consider random deployment of sensor nodes in a two-dimensional square field under free space propagation. The targets to be covered also exist within the same field. Both the sensor nodes and the targets are static, and the sensor nodes are equipped with omni-directional antennas. There is a Base Station (BS) which acts as a sink and collects the data from the various nodes. Each sensor node has two characteristic distances associated with it, namely a sensing range (r) and a one-hop communication range (R), where the former is the distance within which a node can detect the occurrence of any event while the latter is the distance within which it can communicate with other nodes.

A set of sensors exist which have one or more targets within their sensing range. These sensor nodes constitute a set referred to as “Sensing Nodes”, denoted as *S*. These nodes are responsible for sensing any event that occurs at the targets and reporting the same. The rest of nodes, i.e. those nodes which are a part of the WSN and aren’t members of set *S*, are collectively referred to as “Data Forwarding Nodes”, denoted as *D*. These nodes are responsible for forwarding the data packets from the nodes responsible for covering the targets, to the BS. All the nodes can exist in two states: SLEEP state and ACTIVE state. The energy consumption of sensor nodes in the SLEEP state is much less than the consumption in the ACTIVE state, and a significant energy saving can be achieved if the sleep state is employed during the periods of inactivity. We have used an energy model similar to the one described in [40].

B. The Cover-set Scheduling Problem (CSP)

A cover is essentially used to refer to a subset of sensors used at any point of time to cover a subset of targets. The main goal while generating build covers may be to maximize one of the following, if not both – (i) Network Lifetime (MLNB) (ii) Covering Breach (MCBB). The breach is the rate of uncovered targets at any time with Breach rate = 0 and Breach rate = 1 refers to the situations where all targets are covered along the network lifetime and no target is covered along the network lifetime, respectively. The parameters given in the CSP are listed down in Table No. 1. The objective is to find {start_time$_i$} – The start time of the ith cover set where i $\in \{1, 2, ..., C\}$, such that the longest period that a target is uncovered, i.e. the breach θ_{max}, is minimum.

TABLE I. THE PARAMETERS OF THE COVER SET SCHEDULING PROBLEM (CSP)



M	Number of targets
C	Number of covers
S_i	i^{th} cover set
P_i	Duration of cover set Si
T_i	Number of targets covered by Si
U_k	Set of cover sets covering target k

The WSN – CSP is NP-hard in the strong sense and the proof of the same can be found in [35]. All known exact algorithms for these problems require times that grow exponentially with the sizes of problem instances, so for large instances, these algorithms are infeasible. Hence the WSN – CSP is a perfect fit for optimization using the evolutionary computing paradigm. In this paper we have proposed a novel heuristic abbreviated as CSGH for generating the cover sets and have used the HDDE algorithm proposed in [41] for finding out the optimal schedule of these cover sets so as to ensure that θ_{max} is minimum. The HDDE essentially treats the above problem as a general combinatorial optimization problem where its aim is to allot the cover sets to the appropriate time slots with the objective being to minimize θ_{max}, thereby ensuring efficient coverage of targets by the sensor nodes.

IV. THE PROPOSED ALGORITHM : EEFSSP

The EEFSSP goes through a cycle of stages. The key components of the proposed EEFSSP scheme are**:**

A) The Cover Set Generator Heuristic

The proposed Cover Set Generator Heuristic (CSGH) is used to generate cover sets which shall take turns in tracking the given set targets while the other sensing nodes enter the SLEEP state. Hence, each cover set is essentially a subset of the set of sensing nodes in the WSN. The CSGH operates in a manner such that each obtained cover set has an upper limit (N^{max}) set upon its size. This heuristic operates iteratively on the list of sensing nodes, obtaining a cover set from the same each time, and then again operating on the truncated list to obtain further cover sets till the list isn't empty. The obtained cover sets occupy slots within a complete data-sensing round. Thus, this heuristic is applied only once at the very beginning of each such round. The CSGH adopts a *Utility Factor* in order to determine which sensor node it should accept while forming a cover set. The *Utility Factor* for a sensor node may be defined as the ratio of the number of targets that may be covered by including a particular node to the remaining energy of the node. Hence, the smaller the value of the *Utility Factor*, the more is the efficiency of the node. However only the first node in a cover set is selected based upon the afore-mentioned criteria. The remaining nodes are selected based upon whether the respective values of their *Utility Factors* are less than a certain bound or not, with the bound value becoming less rigid as more and more elements are added to the set. This is done so as to ensure that the nodes are distributed equitably among the various covers and that all the sensor nodes with higher sensing efficiency are clustered into a single set which may result in some sets having very minimal sensing efficiency. The algorithm for a single pass of the CSGH has been described in Fig. No. 1.

```
Procedure CSGH

k = 0;
For each sensor node 'i' $\in$ S
do

    Step 1: Calculate the number of uncovered targets covered
    by the sensor node 'i' and store it as $C_i$

    Step 2: Calculate utility factor of the sensor node 'i'
    as $f_i = \frac{C_i}{E_i}$ ($E_i$ is the remaining energy of sensor 'i'). Store
    the value of $f_1$ in F.

    Step 3: If($f_i$ < F)
            F = $f_i$;
            index = i;
            End - If
End - do

Add node 'index' to cover_set
Remove node 'index' from S
F = $C_{index}$;
k = k + 1;
while(k < $N^{max}$ & S $\neq \phi$)

// $N^{max}$ - Maximum number of nodes in each cover set

    For each sensor node 'i' $\in$ S
    do

        Step 1: Calculate the number of uncovered targets covered
        by the sensor node 'i' and store it as $C_i$

        Step 2: Calculate utility factor of the sensor node 'i'
        as $f_i = \frac{C_i}{E_i}$ ($E_i$ is the remaining energy of sensor 'i').

        Step 3: If ($C_i$ < C*(1 + µ))
                Add node 'i' to cover_set
                k = k + 1;
                Remove node 'i' from S
                µ = µ + Δ;
                C = $C_i$;
                End - If

    End - do
End - while
End - CSGH
```

Fig. No. 1 Pseudo-code for the CSGH heuristic

B) A Hybrid Discrete Differential Evolution algorithm (HDDE) to solve the CSP

As the CSP is NP-hard in the strong sense, we have adopted an evolutionary approach to solve it efficiently. To solve the CSP and thereby schedule the cover sets in an efficient manner, we have utilized the Hybrid Discrete Differential Evolution Algorithm (HDDE) algorithm proposed in [41]. The key components of the HDDE are discussed below.

1) Representation of the Schedule (Chromosome)

The proper representation of a solution plays an important role in any evolutionary algorithm. In our paper, a string of positive integers (chromosome) is used to represent a solution. Each element of a chromosome represents a cover set, i.e. a gene is essentially the index of a cover set. For example, Chromosome X represents a permutation where six cover sets are present with are present in sequential order with the cover set having the smallest index occur first and the cover set having the largest index occurring last. Other permutations of a similar form are possible too.

Chromosome X $\longrightarrow$ (1 2 3 4 5 6)

2) The Objective and Fitness Function

The fitness value for any chromosome may be computed as the inverse of θ_{max}, where θ_{max} is referred to as the maximum breach – interval and it represents the longest time for which a target in the WSN remains uncovered. Our objective is to minimize this maximum breach – interval. For any chromosome (representing the cover set schedule) the covers may be generated by utilizing the CSGH, as proposed in Section IV-A. After this, we obtain the value of the fitness function using Equation No. (1). The fitness function is represented by $f_{csp}(V)$ for chromosome *V.*

$$f_{csp}(V) = \frac{1}{\theta_{\max}(V)} \tag{1}$$

3) The Cross-over Operator

The Cyclic Crossover Operator is of immense importance in the proposed HDDE because this operation is carried out at several points in the algorithm. This operator is unique one since it preserves characteristics of both the parents in the offspring. The crossover mechanism may be envisaged via the following example: Let us consider two flow sequences A and B, where A= (1 3 5 6 4 2) and B= (5 6 1 2 3 4). Let the first offspring begin with 1(the starting operation of parent A). Selection of '1' from A implies that '5' should be selected from B, because we want each operation to be derived from one of the two parents. Hence, C= 1 _ 5 _ _ _. This process continues on till after the selection and subsequent insertion of an operation from one of the two parents, the operation in the corresponding position in the other parent is already present in the offspring. After this, the remaining operations are filled in, as per their orders respective to one another in the other string. As can be seen in case of C, insertion of '5' implies that '1' should be inserted in the list, but as '1' is already present in the list (i.e. the starting operation), the cycle stops (hence, the name Cyclic Crossover) and the remaining operations are filled in from B. Hence, C = (1 6 5 2 3 4). Similarly, considering '5' as the starting operation, another offspring D can be obtained in a similar fashion. Hence, D= (5 3 1 6 4 2) .As our algorithm imposes a restriction that the result of each crossover operation results in only a single offspring, the fitter progeny is selected.

4) Perturbation Operator

The HDDE utilizes the Simple Inversion Mutation as the perturbation operator. Two random cut-points are selected in the chromosome and that part of the chromosome between these two cut-points is inverted and placed in the original chromosome. For example, let us consider the chromosome represented by **(ABDEGCFJIH).** Suppose the first cut-point is between D and E and the second cut-point is between F and J. Then the part of the chromosome between the two cut-points is reversed and then placed in the original chromosome to obtain **(ABDFCGEJIH).**

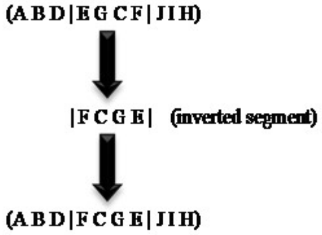


Fig. No. 2 Simple Inversion Mutation Operation

Like other algorithms of the same class, DE is population-based and serves as a stochastic global optimizer. In the classic DE, candidate solutions are represented by a vector of floating point solutions. The mutation process entails the addition of the weighted difference of two members of the population to another third member of the same population to generate a mutated solution. Then, a cross-over operation occurs so as to generate a trial solution that is in turn compared with the target solution using a selection operator, which determines which of the two shall survive for the next generation. It is obvious that traditional DE equations shall not hold in this regard as they had been developed for continuous numerical optimization. As per the novel DDE algorithm proposed by Pan et. al. [42], the target individual is represented by a chromosome of type B (permutation of numbers). The mutant individual is obtained by perturbing the previous generation best solution in the target population. Hence, we achieve the difference variations by applying a perturbation (mutation) operator on the best solution present in the previous generation of the target population. These perturbations are achieved by applying the Simple Inversion Mutation operator on the best solution of the previous generation. Since these perturbations are stochastically different, each result is expected to be distinct from the other. To obtain the mutant, the following equation may be used:

$$V_i^t = \begin{Bmatrix} F(X_g^{t-1})..if..r \le mutation_{prob} \\ X_g^{t-1}..otherwise \end{Bmatrix} \tag{2}$$

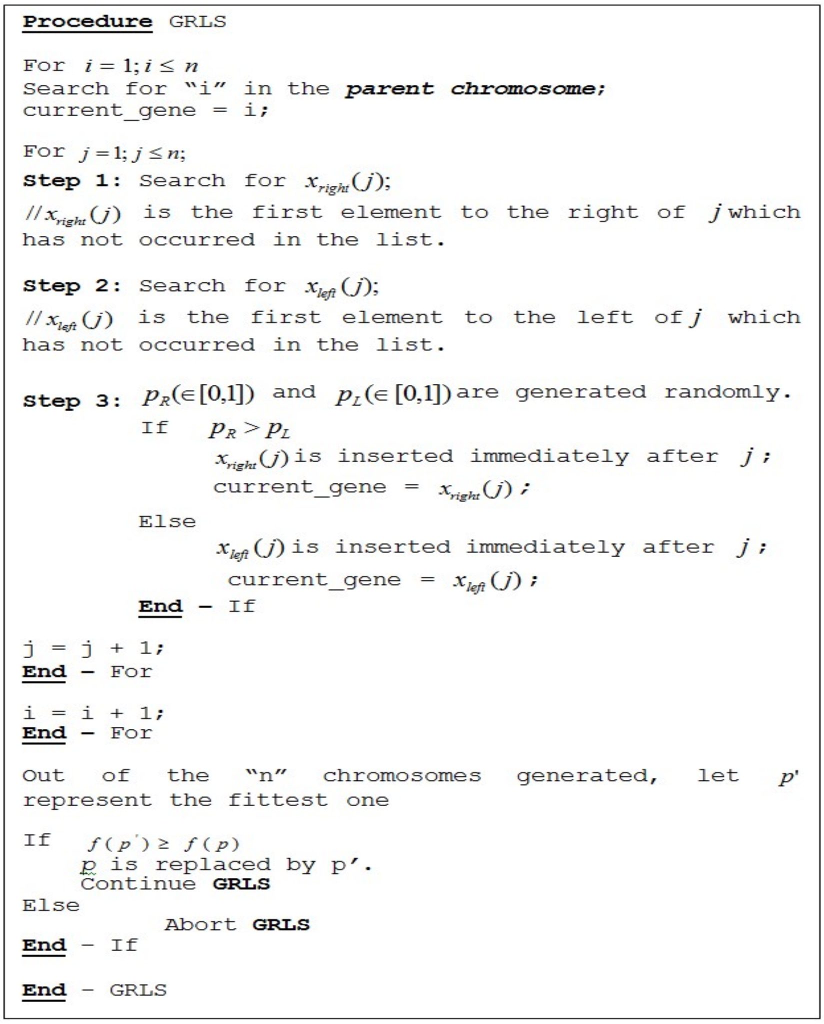
where, X_g^{t-1} represents the best solution of the previous generation of the target population. $F()$ denotes the perturbation operator(Simple Inversion Mutation). Hence, a random number $r \in (0,1)$ is generated and if it is less than $mutation_{prob}$, then the trial solution is obtained by applying the perturbation operator on X_g^{t-1} else, the mutant is set to X_g^{t-1}. After this, we enter the recombination (cross-over) phase. In this, $V^t{}_i$ is crossed over with X_i^t to obtain the trial solution U_i^t if $v \in (0,1)$ [generated randomly] is greater than $crossover_{prob}$. Else, U_i^t is taken to be equal to V_i^t . The pseudo-code representation for the afore-said is:

$$U_i^t = \left\{ \begin{matrix} CR(X_i^{t-1}, V_i^t ..if..v > crossover_{prob} \\ V_i^t ..otherwise \end{matrix} \right\} \quad \textbf{(3)}$$

Hence, if U_i^t is fitter than X_i^{t-1} , then $X_i^t = U_i^t$. Else, $X_i^t = X_i^{t-1}$. Over here, $CR()$ refers to the cross-over operator. As discussed earlier, the Cyclic Cross– Over operator has been used in the HDDE. As it is pretty ostensible, our basic aim is to take advantage of the best solution obtained from the previous generation during the entire search process. Unlike its continuous counterpart, the differential evolution is achieved by the stochastic re-ordering of the previous Hence, if U_i^t is fitter than X_i^{t-1} , then $X_i^t = U_i^t$.

5) Greedy Reordering Local Search (GRLS)

The HDDE had proposed a novel local search operator which restructures a chromosome iteratively on a probabilistic basis to search for fitter solutions. This re-ordering process goes on till the solutions which are obtained from reordering are better in comparison to the parent solution. The GRLS algorithm for a chromosome *p* is given in Fig. No. 2.



```
Procedure GRLS

For $i = 1; i \le n$
Search for "i" in the parent chromosome;
current_gene = i;

For $j = 1; j \le n$;
Step 1: Search for $x_{right}(j)$;
//$x_{right}(j)$ is the first element to the right of $j$ which
has not occurred in the list.

Step 2: Search for $x_{left}(j)$;
//$x_{left}(j)$ is the first element to the left of $j$ which
has not occurred in the list.

Step 3: $p_R(\in[0,1])$ and $p_L(\in[0,1])$ are generated randomly.
        If  $p_R > p_L$
            $x_{right}(j)$ is inserted immediately after $j$;
            current_gene = $x_{right}(j)$;
        Else
            $x_{left}(j)$ is inserted immediately after $j$;
             current_gene = $x_{left}(j)$;
        End - If

j = j + 1;
End - For

i = i + 1;
End - For

Out of the "n" chromosomes generated, let $p'$
represent the fittest one

If   $f(p') \ge f(p)$
    p is replaced by p'.
    Continue GRLS
Else
        Abort GRLS
End - If

End - GRLS
```

Fig. No. 3. Pseudo-code for the GRLS operator

```
Procedure Hybrid Discrete Differential Evolution(HDDE)
Initialize parameters
Initialize target population
Evaluate target population
Apply local search(Greedy Reordering Operator)
generation=0;
/* G is defined by the user */
while (generation<G)
Step 1: Obtain the mutant population
Step 2: Obtain the trial population
Step 3: Evaluate the trial population
Step 4: Select the chromosomes that may progress to
the next generation
Step 6: Apply Local Search(GRLS operator)
Step 7: generation=generation+1;
End - while
Return the best found solution
End - HDDE
```

Fig. No. 4. Pseudo-code for the overall HDDE scheme

C) The Distributed Fuzzy Probability-based Routing Protocol (DFPRP)

In this section, we propose the Distributed Fuzzy Probability-based Routing Protocol (DFPRP) which establishes a set of optimal routes via a set of RNs from the each SN to the BS. The DFPRP is essentially a distributed protocol, which operates like the OLSRP [43], and yields a set of optimal routes which may be used to forward the packets sensed from the targets by each SN by taking the following factors into account**:**

- Distance of the RN from the BS
- Distance of the RN from the node at the previous hop position in the route (the corresponding SN or another RN)
- Residual Energy level of the SN
- Queue Utilization of the SN (indicative of the node congestion level)

Hence, by taking all the afore-mentioned parameters into account the FPRP aims to establish optimal energy, low-congestion and optimal-length routes from each SN to the BS so as to enhance the network lifetime and foster a high packet reception ratio. The optimal route is established on the basis of a Node Probability (NP) which is extracted by the fuzzy model deployed by the routing protocol. During data packet transmission, the source node selects the node with highest NP towards the BS, within its one hop neighbor. In the next hop, the current source node also selects the node with highest NP in its radio range. In this way, hop by hop data routing is obtained from source to BS.

1) Computation of Node Probability

The Node Probability for a node is established on the basis of the following parameters**:**

a) Visibility Metric

The visibility metric (VM) takes into account two factors:

- Distance of the RN from the BS
- Distance of the RN from the node at the previous hop position in the route (the corresponding SN or another RN)

If *d* denotes the distance between the current node and the prospective next node, *d'* denotes the distance of the current node from the BS and *d''* denotes the distance of the prospective next node from the BS, then the Visibility Metric (VM) may be defined as:

$$VM = \frac{(R-d)(d'-d''+R)}{2R^2} \quad (4)$$

We essentially desire to select a node which is neither too close to the BS nor is too close to the current node. Hence to strike a balance and select a node placed at an optimal location we use the VM since it takes care of both the objectives. The higher the value of VM, the more favorable is the node and vice versa.

b) Energy Metric (EM)

The energy metric takes into account the residual energy of a node, whereby it favours those nodes which have a great amount of residual energy because this ensures that the number of packets drops is minimized and the therefore secure routing is achieved. If $E_{residual}$ denotes the residual energy of a node and E_o refers to the initial energy of each node in the WSN, then the Energy Metric (EM) may be defined as**:**

$$EM = \frac{E_{residual}}{E_o} \quad (5)$$

Hence, a node with a higher EM is favoured and vice versa, since the value of EM is directly proportional to the residual energy of the node.

c) Queue Utilization Metric (QUM)

The amount of empty space in a node's queue signifies the level of congestion in that node. Hence, a less congested node has more space in its queue and vice versa. Hence, if Q' refers to the free space in a queue and Q_o refers to the original queue length, then the Queue Utilization Metric (QUM) may be defined as:

$$QUM = 1 - \frac{Q'}{Q_o} \tag{6}$$

Hence, a high value of QUM is favorable and vice versa, since a high value signifies a low degree of congestion in the node, therefore, a lesser chance for buffer overflow, increased latency, packet drops, wastage of energy, deterioration of QoS and lower network throughput.

2) Fuzzy Inferencing of Node Probability

The three parameters discussed in the previous section are inputs to a fuzzy logic controller which estimates the value of the node probability for a certain node on the basis of a rule base followed by a defuzzification mechanism. The general architecture of the FLC is shown in Fig. No. 5 . It consists of four components, namely Fuzzifier, IF-THEN rule base, Fuzzy inference mechanism and Defuzzifier. The Fuzzifier converts crisp input data to fuzzy sets. The Fuzzy output is obtained from fuzzy inference mechanism by combining fuzzy rules into a mapping routine from input to output of the system. Finally, the Defuzzifier extracts a crisp output value from the output fuzzy set. In the proposed protocol we have utilized triangular functions to describe all the fuzzy variables and have adopted the Mamdani method for defuzzification.

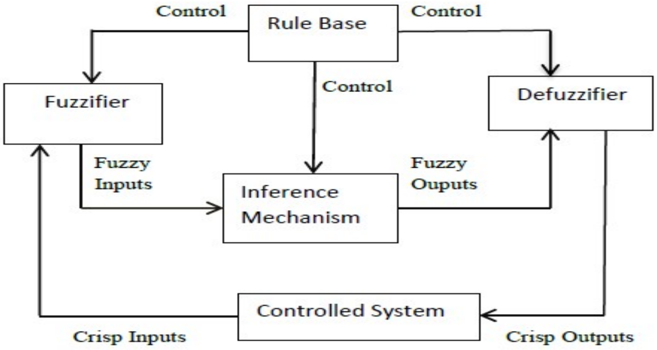


Fig. No. 5. Fuzzy Logic Controller

Each crisp input parameter is represented by an equivalent fuzzy variable, which can be classified into five categories, denoted by assigning appropriate suffixes to each fuzzy variable. These five categories have been listed down in Table No. 2. The crisp values of the three input metrics, the output node probability and their corresponding fuzzy variables have been represented in Table No. 3. The fuzzy rule base for inferencing the appropriate fuzzy variable for the node probability by virtue of the three input metrics has been represented in Table No. 4. The variation of the output Node Potential Metric (NPM) with two input metrics, EM and QUM, is represented in Fig. No. 6.

TABLE II. THE FIVE CATEGORIES OF THE SUFFIXES ASSIGNED TO THE FUZZY VARIABLES

Suffix (Abbreviation)	**Full Form**
VL	Very Low
L	Low
M	Medium
H	High
VH	Very High

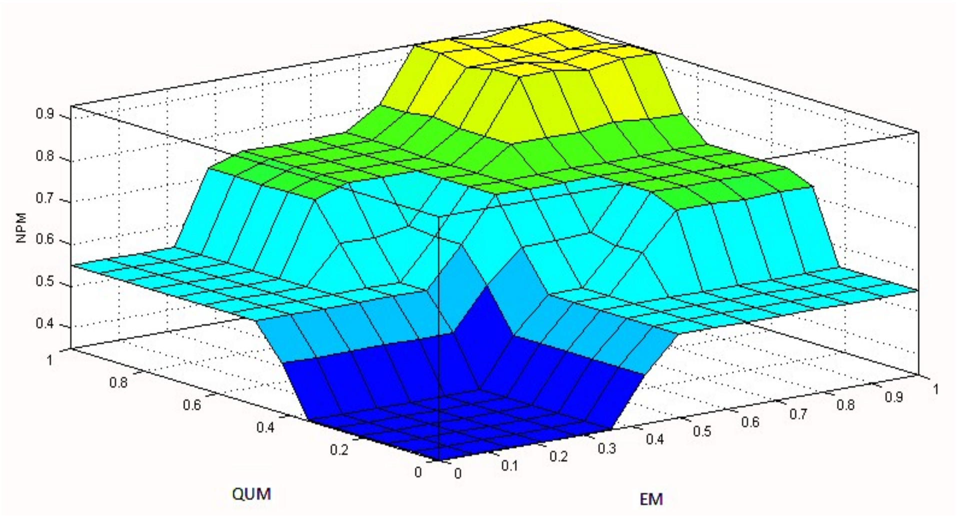


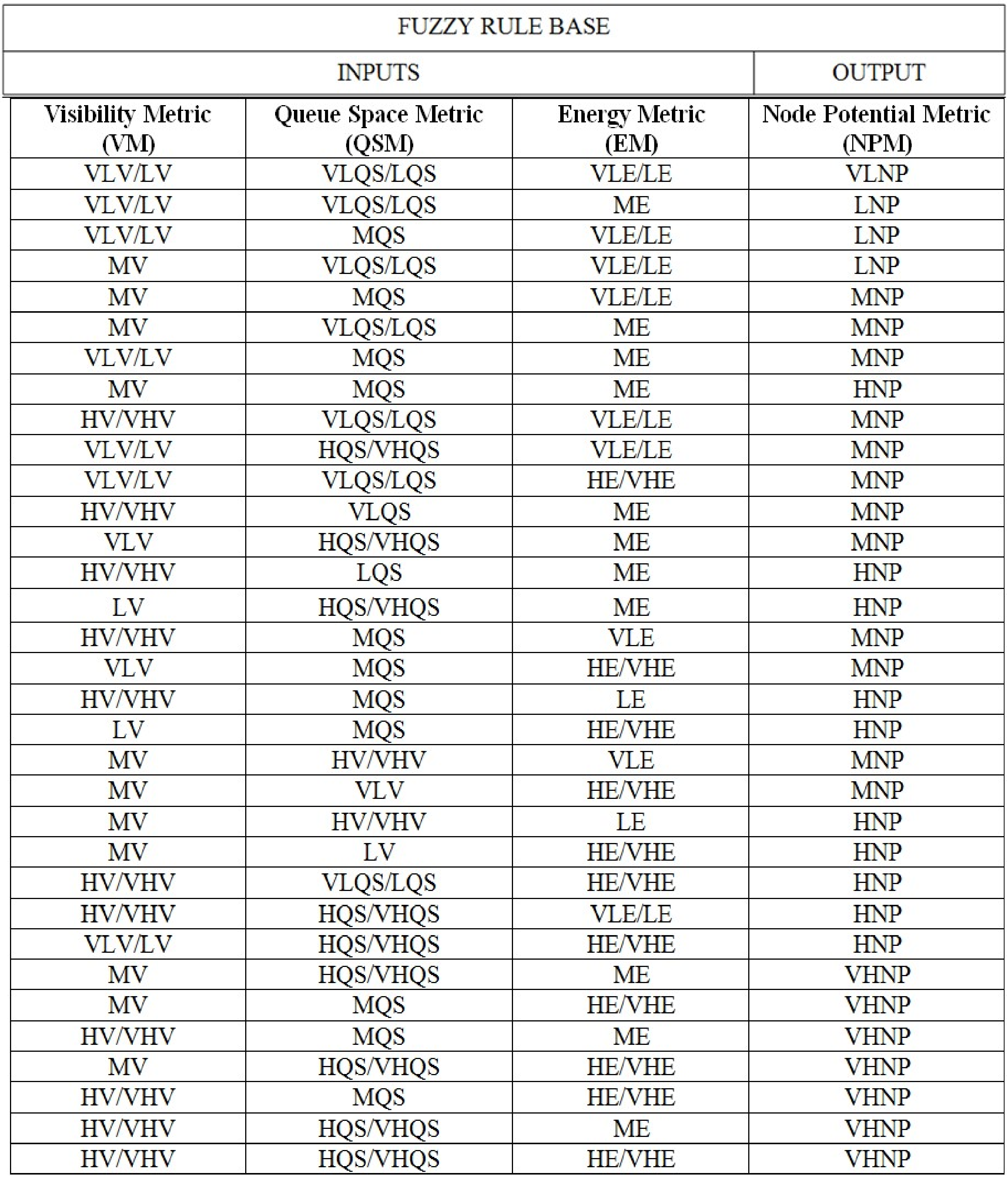
Fig. No. 6. Variation of NPM with variations in QUM and EM (VM-constant)

TABLE III. THE FUZZY VARIABLES AND THEIR CORRESPONDING CRISP RANGES

Crisp Range	**Visibility Metric (VM)**	**Queue Space Metric (QSM)**	**Energy Metric (EM)**	**Node Potential Metric (NPM)**

0 – 0.3	VLV	VLQS	VLE	VLNP
0.25 – 0.45	LV	LQS	LE	LNP
0.4 – 0.7	MV	MQS	ME	MNP
0.6 – 0.9	HV	HQS	HE	HNP
0.8 – 1.0	VHV	VHQS	VHE	VHNP

TABLE IV. THE FUZZY RULE BASE



FUZZY RULE BASE			
INPUTS			OUTPUT
Visibility Metric (VM)	Queue Space Metric (QSM)	Energy Metric (EM)	Node Potential Metric (NPM)
VLV/LV	VLQS/LQS	VLE/LE	VLNP
VLV/LV	VLQS/LQS	ME	LNP
VLV/LV	MQS	VLE/LE	LNP
MV	VLQS/LQS	VLE/LE	LNP
MV	MQS	VLE/LE	MNP
MV	VLQS/LQS	ME	MNP
VLV/LV	MQS	ME	MNP
MV	MQS	ME	HNP
HV/VHV	VLQS/LQS	VLE/LE	MNP
VLV/LV	HQS/VHQS	VLE/LE	MNP
VLV/LV	VLQS/LQS	HE/VHE	MNP
HV/VHV	VLQS	ME	MNP
VLV	HQS/VHQS	ME	MNP
HV/VHV	LQS	ME	HNP
LV	HQS/VHQS	ME	HNP
HV/VHV	MQS	VLE	MNP
VLV	MQS	HE/VHE	MNP
HV/VHV	MQS	LE	HNP
LV	MQS	HE/VHE	HNP
MV	HV/VHV	VLE	MNP
MV	VLV	HE/VHE	MNP
MV	HV/VHV	LE	HNP
MV	LV	HE/VHE	HNP
HV/VHV	VLQS/LQS	HE/VHE	HNP
HV/VHV	HQS/VHQS	VLE/LE	HNP
VLV/LV	HQS/VHQS	HE/VHE	HNP
MV	HQS/VHQS	ME	VHNP
MV	MQS	HE/VHE	VHNP
HV/VHV	MQS	ME	VHNP
MV	HQS/VHQS	HE/VHE	VHNP
HV/VHV	MQS	HE/VHE	VHNP
HV/VHV	HQS/VHQS	ME	VHNP
HV/VHV	HQS/VHQS	HE/VHE	VHNP

3) *Overall Routing Protocol*

As shown in Fig. No. 7, the DFPRP is a hop-by-hop routing protocol similar to the OLSRP where at each hop the current node determines which node ought to be the next node based on a set of criteria. The DFPRP routine returns a set of optimal routes with a route probability associated with each. The route which isn't under use and has the highest route probability is selected to transmit data packets to the BS. Thus, the overall protocol for the Distributed Fuzzy Probability-based Routing Protocol (DFPRP) may be represented as shown in Fig. No. 8.

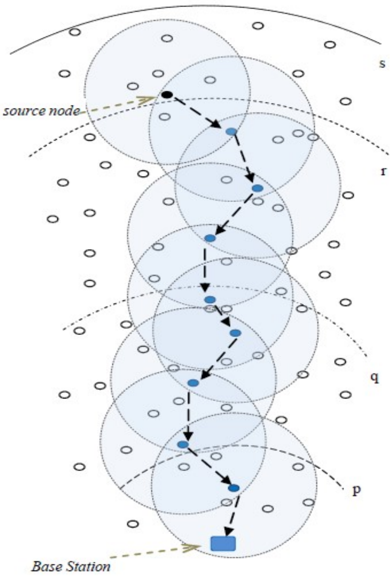


Fig. No. 7. An example of hop-by-hop routing in WSNs

```
Procedure DFPRP
For each Routing Node i which is a one-hop neighbor of node D
do
current_node = i;
k=1;
p=1;
route_list$_i$(k)=current_node;
while (current_node is not a one-hop neighbor of BS)
Step 1: For each Routing Node i which is a one-hop neighbor of current_node, compute the probability of packet transmission (denoted by $P_i$).
Step 2: Find the node i for which $P_i$ is maximum.
        current_node = i;
        k = k + 1;
        route_list$_i$(k)=current_node;
Step 3: p = p * $p_i$;
End - while
route_probability$_i$ = $(p)^{(1/k)}$;
End - do

End - DFPRP
```

Fig. No. 8. Pseudo-code for the DFPRP

D) Overall Scheme for the EEFSSP

```
Procedure EFSSP

nr = 1;
while (nr ≤ NR)//NR - Maximum number of rounds
Determination of an optimal schedule:

Step 1: Select each sensing node 'i' such that Ei ≥ Eth and
        store it in set S

Step 2: ns=0; // To store the number of slots
        while ($S \neq \phi$)
        ns = ns+1;
        ($CS_{ns}$,S) = CSGH (S,N);
       // N - Maximum size of each cover set
        End - while

Step 3: Find the optimal schedule of the obtained cover sets
        where each cover set occupies a slot in a round.
       (ns - Total number of slots)using the stated HDDE
        scheme
Data Routing Phase:

For each slot 'S'-

Step 1: All the sensing nodes in CS enter the ACTIVE state
        while all those nodes (sensing/routing) awake in the
        previous slot enter the SLEEP state if they do not
        have any outstanding packets. In case they do have
        outstanding packets, they stay in the ACTIVE state
        σ number of slots or till the end of the current
        round (whichever occurs earlier) and drop the
        residual packets(if any)

Step 2: The route with highest probability for packet
        transmission is determined using the DFPRP. The
        nodes that constitute the route enter the ACTIVE
        state (unless already ACTIVE) and the packets are
        routed along the determined route to the BS. In case
        no such route is found, then the node
         holds the packets as per the policy described in
        the previous step

Step 3: The energies of all the sensor nodes are updated

  nr=nr+1;
End - EFSSP
```

Fig. No. 9. Pseudo-code for the overall EEFSSP

V. RESULTS

In our tests on the various problem instances, the proposed EEFSSP algorithm is coded in MATLAB and the experiments are executed on a Pentium P-IV 3.0 GHz PC with 512MB memory. To verify the efficacy of the proposed protocol, we have performed three sets of tests. In the first set of tests, the performance of the HDDE algorithm is compared vis-a-vis another relevant evolutionary algorithm proposed in [44], for the Cover-Set Scheduling Problem (CSP), over several runs on various instances. The second set of tests comprise of extensive simulations to compare the performance of the DFPRP against other relevant routing protocols proposed in [11] and [13], where the criteria under consideration are energy efficiency (network lifetime) and reliability (packet reception ratio). Finally, exhaustive comparisons of the performance of the overall EEFSSP with other stationary target coverage protocols described in [36] and [39] respectively, are made, with the metric for comparison being the network lifetime.

A) Performance of the HDDE algorithm (CSP)

720 instances are solved for the MNLB problem using a column generator heuristic [44]. The corresponding parameters are listed down in Table No. 5. For each combination, 30 instances are generated.

TABLE V. THE FUZZY RULE BASE

Abbreviation	Meaning
N	Number of sensors
M	Number of targets
W	Bandwidth Constraint
α	Breach Rate
Q'	Average number of covers

The performance of the HDDE algorithm as a cover-set scheduler has been compared with the CSGA proposed in [44]. Both the algorithms have been tested on a number of instances generated for α=0.1 and α=0.2, respectively. It's quite evident from Table No. 6 and Table No. 7, that the HDDE algorithm clearly outperforms the CSGA. The efficiency of the HDDE algorithm compared to the CSGA is especially accentuated for α=0.2.

TABLE VI. COMPARISON OF THE PERFORMANCES OF THE HDDE AND THE CSGA ON THE INSTANCES FOR α=0.1

Parameter	**α=0.1**				
	% HDDE	**#HDDE**	**%CSGA**	**#CSGA**	**Improvement**
M=50,W=5,Q'=41	5.3	26	10.8	21	4.96
M=50,W=10,Q'=47	5.6	27	5.8	25	0.26
M=50,W=50,Q'=47	4.7	27	8.8	23	3.76
M=100,W=5,Q'=90	1.2	28	2.1	27	0.88
M=100,W=10,Q'=96	0.1	29	1.1	28	0.98
M=100,W=50,Q'=96	0	30	0.3	29	0.3
M=150,W=5,Q'=137	0.8	29	1.4	28	0.59
M=150,W=10,Q'=142	0.4	29	1.6	28	1.18
M=150,W=50,Q'=146	0	30	0.9	29	0.89
M=200,W=5,Q'=182	1.3	28	3.4	26	2.07
M=200,W=10,Q'=188	0	30	0.5	28	0.49
M=200,W=50,Q'=189	0.8	29	1.8	27	1.01
Total No. of opt found		**342**		**319**	
Average Deviation from LB	**1.68%**		**3.21%**		
Improvement over CSGA					**1.45%**

TABLE VII. COMPARISON OF THE PERFORMANCES OF THE HDDE AND THE CSGA ON THE INSTANCES FOR α=0.2

Parameter	**α=0.2**				
	% HDDE	**#HDDE**	**%CSGA**	**#CSGA**	**Improvement**
M=50,W=5,Q'=41	9.4	13	13.44	4	3.56
M=50,W=10,Q'=47	10.2	15	15.22	10	4.31
M=50,W=50,Q'=47	12.3	13	18.98	2	5.62
M=100,W=5,Q'=90	18.44	8	27.87	0	7.01
M=100,W=10,Q'=96	9.32	10	14.32	5	4.37
M=100,W=50,Q'=96	7.34	12	11.22	10	3.49
M=150,W=5,Q'=137	8.54	7	12.18	8	3.25
M=150,W=10,Q'=142	11.43	2	21.34	1	8.19
M=150,W=50,Q'=146	8.92	11	14.24	12	4.66
M=200,W=5,Q'=182	13.2	13	20.22	0	6.12
M=200,W=10,Q'=188	9.23	6	16.72	3	6.34
M=200,W=50,Q'=189	7.55	8	15.35	7	6.95
Total No. of opt found		**118**		**62**	
Average Deviation from LB	**10.49%**		**16.76%**		
Improvement over CSGA					**5.33%**

B) Performance of the DFPRP

The merits of the proposed DFPRP have been investigated by means of extensive MATLAB simulations. We have considered an arbitrary architecture where a number of sensor nodes have been deployed randomly into a field of dimension 200 m * 200 m. The distance of each node from the BS is taken as constant throughout the simulation. The routing protocol takes into account three metrics, namely visibility metric, energy metric and queue utilization metric where the factors taken into consideration are distance of a node with respect to other nodes & the BS (VM), the energy left in the battery powering the node (EM) and the empty space left in the queue of a node (QUM), respectively. These three factors have been taken into consideration to ensure that the protocol is energy-efficient and ensures high packet reception ratio by minimizing the number of packet drops. Based on these three metrics, each node obtains a probability of which node ought to follow it in the normal routing path until the node forwarding data packets is at a one-hop distance from the BS. We have utilized a fuzzy inferencing scheme for computing this probability. Quite intuitively, several different paths are adopted in order to transmit data packets to the BS as the value of NP for the various nodes, changes regularly.

The comparison of the DFPRP is made with two algorithms, namely LEACH and MRP for different initial node energies on a 200 m * 200 m WSN field. The number of rounds verses percentage of dead nodes for the aforementioned protocols are given in Table No. 8, for various initial node energies. The simulation and experimental results indicate that the DFPRP provides higher network lifetime, at the various node energies, compared to other similar protocols and thereby outperforms its peers, especially when the initial node energy is 1.0 Joules.

TABLE VIII. NUMBER OF ROUNDS VERSUS PERCENTAGE OF DEAD NODES FOR THE DIFFERENT PROTOCOLS AT DIFFERENT INITIAL ENERGIES

Initial Energy (J/Node)		**Protocol**	**Percentage of Dead Nodes**						
			1%	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**
0.25	**Number Of Rounds**	**LEACH**	411	505	711	765	816	855	910
		MRP	478	500	699	772	826	878	905
		DFPRP	590	620	750	832	906	997	1024
0.5		**LEACH**	782	841	913	1011	1211	1253	1319
		MRP	806	866	932	1058	1266	1297	1333
		DFPRP	908	992	1058	1293	1342	1409	1469
1.0		**LEACH**	1569	1732	1943	2196	2301	2510	2797
		MRP	1402	1615	1792	1912	2066	2215	2403
		DFPRP	1611	1828	2025	2316	2488	2602	2893

We have also performed simulations to verify how the packet reception ratio (PRR) changes with respect to the maximum number of retransmission attempts (MRA) for the DFPRP vis-a-vis LEACH and MRP. The PRR is calculated as the ratio of the number of packets received successfully to the total number of packets transmitted. The packet retransmission is required in case of unsuccessful packet delivery. In our experiment, we calculate the fraction of the packets reaching the BS successfully, by varying the number of retransmission attempts. As the number of retransmission attempts increase, the PRR also increases. In case of 100% successful packet delivery, PRR is equal to one. . Table No. 9 represents the maximum and minimum number of packets delivered successfully to the BS for different values of MRA for LEACH, MRP and DFPRP algorithms respectively. MIN and MAX refer to the maximum and minimum values of PRR obtained over 30 runs for transmitting 25 packets.

TABLE IX. PRR VALUES FOR THE THREE PROTOCOLS

Maximum Number of Retransmission Attempts	**LEACH**		**MRP**		**DFPRP**	
	MIN	**MAX**	**MIN**	**MAX**	**MIN**	**MAX**
1	0.53	0.64	0.58	0.69	0.66	0.81
2	0.61	0.68	0.72	0.88	0.88	1
3	0.74	0.82	0.81	0.98	1	1
4	0.81	0.97	0.85	1	1	1
5	0.96	1	1	1	1	1
6	1	1	1	1	1	1

C) Performance of the EEFSSP

We have observed the performance of the proposed EEFSSP in target coverage by means of extensive simulations in a simulator developed using MATLAB. We have considered a 2-dimensional WSN field comprising of several randomly deployed sensor nodes and the targets that have to be monitored. We have considered a field of dimensions 200 m * 200 m with 50 sensor motes deployed in them randomly in order to detect 10 targets which generate packets in bursts with each burst yielding N' packets, where $N' \in (0, 500)$.

We have compared the performance of our scheme with the target coverage algorithm proposed in [36] (referred to as A_1) and the TC algorithm proposed in [39]. The results obtained from testing the three algorithms on a fixed

WSN architecture for different initial node energies, i.e. 0.25 J/node, 0.5 J/node and 1.0 J/node, respectively are listed in Table No. 10. It is quite clear from the results that the EEFSSP clearly outperforms its TC and A_1. This can be attributed to the efficient strategies that have been incorporated into this scheme such as energy saving by virtue of sleep-scheduling, formation of optimal cover sets and an energy-efficient routing protocol which considers three profound metrics in its ambit to facilitate lower packet drops, lesser overhead and longer network lifetime.

TABLE X. NUMBER OF ROUNDS VERSUS PERCENTAGE OF DEAD NODES FOR THE DIFFERENT PROTOCOLS AT DIFFERENT INITIAL ENERGIES

Initial Energy (J/Node)		**Protocol**	**Percentage of Dead Nodes**						
			1%	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**
0.25	**Number Of Rounds**	**A_1**	206	281	311	342	359	391	412
		TC	272	299	329	366	389	421	444
		EEFSSP	384	513	560	592	616	655	721
0.5		**A_1**	382	441	513	611	715	753	819
		TC	406	466	532	658	766	797	833
		EEFSSP	508	592	658	793	842	909	1002
1.0		**A_1**	833	892	978	1066	1122	1343	1422
		TC	855	1002	1101	1222	1366	1488	1532
		EEFSSP	1001	1221	1332	1444	1523	1621	1801

VI. CONCLUSION

In this paper, we have considered the target coverage problem pertaining to WSNs and have proposed an evo-fuzzy sleep scheduling scheme, EEFSSP, for the same. The proposed scheme divides the set of sensing nodes into cover sets which alternately take turns to monitor the targets while the others are in their low-energy SLEEP states. Firstly, the proposed scheme introduces a novel heuristic for generating cover sets (GSGH) which iteratively gathers nodes to form cover sets so as to ensure that the generated cover sets are equitable in nature. Secondly, the EEFSSP also introduces a novel evolutionary approach (HDDE) to the cover set scheduling problem. Thirdly, the proposed scheme comprises of a novel hop-by-hop routing scheme DFPRP, to transmit the data packets sensed by the nodes to the Base Station. The scheme takes three metrics into account to foster energy efficiency and ensure that the number of packet drops is minimized. The performances of the HDDE, DFPRP and the overall EEFSSP have been tested through extensive simulations and they clearly outperform their peers. The target coverage problem is of profound importance and finds application in a variety of fields. Hence, the proposed scheme can be used in a variety of problems pertaining to WSNs. As a future work, we wish to explore and integrate the concept of trust management into the proposed scheme so as to ensure that the data transfer process is secure. Secondly, we wish to observe how the performance of the overall protocol and the DFPRP in particular will change if we adopt a type-II fuzzy scheme instead of the traditional type-I fuzzy scheme. Thirdly, we wish to test its performance on larger networks comprising of a large number of heterogeneous nodes so as to verify whether the proposed scheme is applicable under all conditions or not. Finally, we desire to test its hardware implementation with Iris motes using TinyOS under various conditions.

REFERENCES

[1] Raghavendra, Cauligi S., Krishna M. Sivalingam, and Taieb Znati, eds. *Wireless sensor networks*. Springer, 2004.

[2] Ye, Wei, John Heidemann, and Deborah Estrin. "Medium access control with coordinated adaptive sleeping for wireless sensor networks." *Networking, IEEE/ACM Transactions on* 12.3 (2004): 493-506.

[3] Lu, Gang, et al. "Delay efficient sleep scheduling in wireless sensor networks."*INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. Vol. 4. IEEE, 2005.

[4] Rossi, André, Alok Singh, and Marc Sevaux. "Column generation algorithm for sensor coverage scheduling under bandwidth constraints." *Networks* 60.3 (2012): 141-154.

[5] Wang, Xiaorui, et al. "Integrated coverage and connectivity configuration in wireless sensor networks." *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003.

[6] Zou, Yi, and Krishnendu Chakrabarty. "A distributed coverage-and connectivity-centric technique for selecting active nodes in wireless sensor networks."*Computers, IEEE Transactions on* 54.8 (2005): 978-991.

[7] Zhang, Honghai, and Jennifer C. Hou. "Maintaining sensing coverage and connectivity in large sensor networks." *Ad Hoc & Sensor Wireless Networks*1.1-2 (2005): 89-124.

[8] Paradis, Lilia, and Qi Han. "A survey of fault management in wireless sensor networks." *Journal of Network and Systems Management* 15.2 (2007): 171-190.

[9] Polastre, Joseph, Jason Hill, and David Culler. "Versatile low power media access for wireless sensor networks." *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004.

[10] Ye, Wei, John Heidemann, and Deborah Estrin. "An energy-efficient MAC protocol for wireless sensor networks." *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 3. IEEE, 2002.

[11] Xiangning, Fan, and Song Yulin. "Improvement on LEACH protocol of wireless sensor network." *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*. IEEE, 2007.

[12] Lindsey, Stephanie, and Cauligi S. Raghavendra. "PEGASIS: Power-efficient gathering in sensor information systems." *Aerospace conference proceedings, 2002. IEEE*. Vol. 3. IEEE, 2002

[13] Jing Yang, Mai Xu, Wei Zhao and Baogue Xu, " A Multipath Routing Protocol Based on Clustering and Ant Colony Optimization for Wireless Sensor Networks", Sensors ISSN 1424-8220, 10,4521-4540, doi : 10.3390/s 100504521' 2010.

[14] Akkaya, Kemal, and Mohamed Younis. "A survey on routing protocols for wireless sensor networks." *Ad hoc networks* 3.3 (2005): 325-349.

[15] Yu, Yan, Ramesh Govindan, and Deborah Estrin. *Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks*. Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.

[16] Heinzelman, Wendi Rabiner, Joanna Kulik, and Hari Balakrishnan. "Adaptive protocols for information dissemination in wireless sensor networks."*Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*. ACM, 1999.

[17] Manjeshwar, Arati, and Dharma P. Agrawal. "TEEN: ARouting Protocol for Enhanced Efficiency in Wireless Sensor Networks." *IPDPS*. Vol. 1. 2001.

[18] Manjeshwar, Arati, and Dharma P. Agrawal. "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks." *ipdps*. Vol. 2. 2002.

[19] Bandyopadhyay, Seema, and Edward J. Coyle. "An energy efficient hierarchical clustering algorithm for wireless sensor networks." *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 3. IEEE, 2003.

[20] Karlof, Chris, and David Wagner. "Secure routing in wireless sensor networks: Attacks and countermeasures." *Ad hoc networks* 1.2 (2003): 293-315.

[21] T. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2003.

[22] Ma, Junchao, et al. "Energy efficient TDMA sleep scheduling in wireless sensor networks." *INFOCOM 2009, IEEE*. IEEE, 2009.

[23] Pantazis, Nikolaos A., et al. "Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling." *Ad Hoc Networks* 7.2 (2009): 322-343.

[24] Subramanian, Ramanan, and Faramarz Fekri. "Sleep scheduling and lifetime maximization in sensor networks: fundamental limits and optimal solutions."*Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006.

[25] Tian, Di, and Nicolas D. Georganas. "A node scheduling scheme for energy conservation in large wireless sensor networks." *Wireless Communications and Mobile Computing* 3.2 (2003): 271-290.

[26] Deng, Jing, et al. "Scheduling sleeping nodes in high density cluster-based sensor networks." *Mobile Networks and Applications* 10.6 (2005): 825-835.

[27] Cardei, Mihaela, et al. "Energy-efficient target coverage in wireless sensor networks." *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*. Vol. 3. IEEE, 2005.

[28] Chakrabarty, K., Iyengar, S., Qi, H., Cho, E.: Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks. IEEE Trans. on Computers 51(12), 1448–1453 (2002)

[29] Dong, Q., Shukla, A., Shrivastava, V., Agrawal, D., Banerjee, S., Kar, K.: Load Balancing in Large-Scale RFID Systems. IEEE INFOCOM, 2281–2285 (2007)

[30] Wang, W., Srinivasan, V., Wang, B., Chua, K.: Coverage for Target Localization in Wireless Sensor Networks. IEEE Trans. on Wireless Communications 7(2), 667–676 (2008)

[31] Cai, Y., Lou, W., Li, M.: Target-Oriented Scheduling in Directional Sensor Networks. IEEE INFOCOM, 1550–1558 (2007)

[32] Lu, M.,Wu, J., Cardei, M., Li, M.: Energy-Efficient Connected Coverage of Discrete Targets in Wireless Sensor Networks. In: Lu, X., Zhao, W. (eds.) ICCNMC 2005. LNCS, vol. 3619, pp. 43–52. Springer, Heidelberg (2005)

[33] Li, D., Cao, J., Liu, M., Zheng, Y.: K-Connected Target Coverage Problem in Wireless Sensor Networks. IEEE INFOCOM, 1976–1984 (2005)

[34] Zhou, Z., Das, S., Gupta, H.: Fault Tolerant Connected Sensor Cover with Variable Sensing and Transmission Ranges. In: IEEE Commumications Society Conference on Sensor and Ad Hoc Communications and Networks, pp. 594–604 (2005)

[35] Rossi, André, Alok Singh, and Marc Sevaux. "Génération de colonnes et réseaux de capteurs sans fil." *Proc. of ROADEF* (2010).

[36] Liu, Yuzhen, and Weifa Liang. "Prolonging network lifetime for target coverage in sensor networks." *Wireless Algorithms, Systems, and Applications*. Springer Berlin Heidelberg, 2008. 212-223.

[37] Kar, K., Kodialam, M., Lakshman, T.V., Tassiulas, L.: Routing for Network Capacity Maximization in Energy-Constrained Ad Hoc Networks. IEEE INFOCOM, 673–681 (2003)

[38] Jaggi, N., Abouzeid, A.: Energy-Efficient Connected Coverage in Wireless Sensor Networks. In: The Forth Asia International Mobile Computing Conf., pp. 77–86 (2006)

[39] Ding, Ling, et al. "Constant-approximation for target coverage problem in wireless sensor networks." *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012.

[40] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," IEEE Transactions on Wireless Communications, vol. 4,no. 5, pp. 2349–2360, 2005

[41] Ganguly, S., Chowdhury, A., Mukherjee, S., Suganthan, P. N., Das, S., & Chua, T. J. (2013). A Hybrid Discrete Differential Evolution Algorithm for Economic Lot Scheduling Problem with Time Variant Lot Sizing. In *Soft Computing Models in Industrial and Environmental Applications* (pp. 1-12). Springer Berlin Heidelberg.

[42] Pan, Quan-Ke, Mehmet Fatih Tasgetiren, and Yun-Chia Liang. "A discrete differential evolution algorithm for the permutation flowshop scheduling problem."*Computers & Industrial Engineering* 55.4 (2008): 795-816.

[43] Medina, E., Meseguer, R., Molina, C., & Royo, D. (2011). OLSRp: Predicting Control Information to Achieve Scalability in OLSR Ad Hoc Networks. In *Mobile Networks and Management* (pp. 225-236). Springer Berlin Heidelberg.

[44] Rossi, A., Sevaux, M., Singh, A., & Geiger, M. J. (2011). On the cover scheduling problem in wireless sensor networks. In *Network Optimization* (pp. 657-668). Springer Berlin Heidelberg.