JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 885-908 (2023)
DOI: 10.6688/JISE.202307_39(4).0012

MECT-PSO: A Reliable and Energy Efficient Topology Control Protocol for Wireless Sensor Networks*

POONAM T. AGARKAR[1], MANISH D. CHAWHAN[2], KISHOR D. KULAT[3] AND PRATIK R. HAJARE[4]
[1,2]*Department of Electronics and Telecommunication Engineering*
Yeshwantrao Chavan College of Engineering
Maharashtra, 441110 India
[3]*Department of Electronics and Telecommunication Engineering*
Visvesvaraya National Institute of Technology
Maharashtra, 440010 India
[4]*Department of Electronics and Telecommunication Engineering*
Mansarovar Global University
Madhya Pradesh, 441108 India
E-mail: poonamagarkar71@gmail.com[1]*; mchawan1001@gmail.com*[2]*;*
kdkulat@ece.vnit.ac.in[3]*; pratik_hajare8@gmail.com*[4]

Recent advancement in wireless sensor networks consider independent layer parameters jointly exploiting existing collaborative mechanism using cross layer design (CLD) optimization maximizing the overall network performance in terms of reduction of energy consumption, efficient routing, and lifetime enhancement. Integrating layer functionalities pose challenges on objective functions relating to complexity and non-linearity making it highly multimodal thus complicating the search boundaries. Due to flexibility and versatility, the computationally intelligent Bio-inspired swarm are very efficient in solving such non-linear design issues. The work introduces a hybrid approach using the minimum execution/completion time (MECT) scheduling and particle swarm optimization (PSO) scheme called MECT-PSO protocol to improve the performance of Routing protocol ensuring an optimal path and then minimizing the average energy loss (AEL) by localizing the intermediate nodes (*IN*'s) along the optimal route. The strategy of shortest path with minimum hop count is exploited by considering alternative routes from source to receiver and an optimal path is chosen based on MECT algorithm considering queue status and the node overheads. The optimal path length is reduced by localizing the *IN*'s to find the best positions under topological constraints using novel swarm approach. The self-organizing approach of nodes provides better connectivity, coverage and reliable paths. Analysis and evaluation of the proposed MECT-PSO protocol implemented using MATLAB platform showed significant improvement over existing algorithms in terms of residual energy, surviving nodes and coverage.

Keywords: cross layer design, bio-inspired swarm, minimum execution/completion time, particle swarm optimization, self-organizing and average energy loss

1. INTRODUCTION

In WSN, energy can be conserved by finding the optimal path for transmission from the source node to the destination. Generally, the path selected by different routing schemes considers few of the quality access parameters and neglect other factors thus reducing the efficiency of the system. Routing scheme considers minimum hop count with some other

Received December 22, 2021; revised May 3, 2022; accepted September 8, 2022.
Communicated by Chun-Hung Richard Lin.
* This paper is the part of DST-SERB funded project titled "Energy and Communication Efficient Protocol of Mobile Adhoc Network using Cross Layer Approach," Grant No. EEQ/2018/000540.

parameter and selects the path without considering other factors such as *IN*'s queue status, congestion, response time *etc.* On the other hand, the scheduling algorithms should be able to ensure maximum element utilization, fair allocation, maximum throughput, minimum task execution time, minimum waiting in queue and minimum response time. Lastly in dynamic networks the nodes are accelerated randomly to occupy new locations in the network area which results in a scattered form of the network after some packet transmissions. The above three factors greatly affect the overall performance of the network with respect to various parameters and greatly affects the network lifetime due to unwanted energy loss from the participating sensor elements. Also, part of energy is consumed in flooding of control information (initially during route searching mechanism, acknowledgements and route maintenance) and changes in network structures (due to node movements, early death of the nodes and link failures). To meet the current network requirement, traditional approach of layer specific protocols is being replaced by protocols considering multiple layers and allowing them to share network information and work cooperatively with unified efforts to use the resources with optimum. The concept termed by researchers a Cross Layer Design (CLD) approach is to utilize the nodes constrained resources effectively and efficiently. A general system employing a cross layer design approach for a routing scheme is shown in Fig. 1 below.

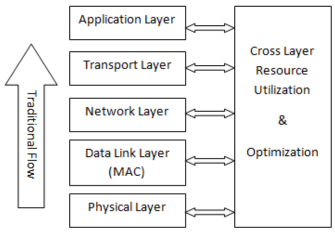


Fig. 1. Cross layer design approach for communication protocol.

The existing routing protocol standardized by IETF considers the least hop count neglecting other important factors such as the energy utilization, link stability and residual energy of the network which adversely affects performance required for any routing protocol. The routing mechanism is basically related to flooding of the request packets to neighboring nodes prove to be inefficient related to finding route and thus provides low packet delivery in dynamic situations, drastic long delays, energy consumption and extra overheads due to routing. This research work focuses on improving the performance of the routing scheme with respect to energy consumption considering overheads to improve the lifetime of the network using a hybrid approach by combining the features of MECT scheduling and PSO algorithm. The MECT algorithm is effective under heterogeneous conditions where nodes have varying capabilities. MECT select the path with minimum time thus reducing the end-to-end delay whereas PSO governs the mobility of the *IN*'s so as to reduce the distance between the source and the destination maintaining their own neighborhood and the minimum distance between the old neighbors. The first stage of the proposed algorithm allows only those nodes to transmit and receive whose individual energy is greater than or equal to the average energy of the network. The combined approach of MECT-PSO provides a second stage energy conservation by optimizing the processing time and energy by reducing the distance between the source and the receiver through optimum placement of *IN*'s along the selected path.

1.1 Problem Statement

Maintaining internode connectivity and network coverage is a crucial concern. The non-uniform activity of sensor nodes in hybrid environment depletes the battery of few nodes quicker than the rest of the network. Frequent use of some high probable nodes may even exhaust the energy forcing them to die before other nodes. The early failure of such nodes in crucial regions may create holes and results in a partitioned network. A typical case of this point is so called the Hot Spot issue. In spite of densely populated network to tolerate node failures, we have to face the problem of isolated sinks caused due to depleted neighbors. This may result in failure of complete network leaving the remaining nodes to be perfectly functional.

The criteria to select forwarding/intermediate nodes basically takes it into account the residual energy and the signal strength. Thus, alternative paths after route discovery mechanism are not considered and neglected. Other possible links having greater hops as compared to the selected link may be reliable in terms of other node overheads.

Another essential task is self-organization of network for optimum operations. Random mobility to sensor nodes causes unpredictable and uncertain network topologies. Therefore, nodes in the network should be stimulated during their lifetime to form a feasible network at any instance. Most of the node energy is consumed during communication. Nodes at distant below transmission range consume energy more and will be prone to early link failure reducing hops. On the other hand, when nodes are close, hop count increases but saves energy and extends link life time.

Keeping in mind all these aspects, our problem statement can be formulated as follows: given a set of nodes,

- How can we maintain uniform residual energy for all nodes at any given time?
- In what way we can prevent early link failures and improve link expiration time?
- How can we reduce the energy consumption by intermediate nodes during communication?
- In what way, an efficient runtime topology be maintained?

1.2 Contributions

To improve the performance of state-of-art routing mechanism, our contribution includes:

1. Node with residual energy greater than the average residual energy of the network is selected as source, forwarder or destination. It ensures that none of the network node will die early with respect to other nodes and thus improves the lifetime of the network.
2. All possible links from source to destination are considered and evaluated in terms of execution/completion time. The best route is selected based on minimum execution/ completion time relaxing the condition of minimum hops. The overheads at each of the intermediate nodes are taken into consideration to find the maximum begin time and

execution/completion time. For the best path, the execution time should be less than the minimum begin time (minimum begin time from all maximum begin times calculated for all paths) or otherwise the best path is selected with respect to minimum completion time. Selecting minimum load path with higher execution time ensures higher, fast and successful packet delivery with long link lifetime.

3. Once an optimum path is selected, the idea is to find new positions for the intermediate nodes along the path to reduce the length of the path. The localization will maintain existing neighbors of the intermediate nodes further restricting the nodes to be in the close vicinity of any other node for better network structure. Swarm intelligence is used for search and exploration in self-organizing networks under such highly constrained environment. The feasible runtime locations after initial deployment is dynamically controlled to avoid populated and scarce regions. The data packets will be sent once the new optimum position for the intermediate nodes are found. The optimum positions will be such to reduce the length of the link thus reducing energy consumption during transmission. After few iterations the randomly deployed nodes will spread the populated nodes over the network space and fills out scarce regions. The network will be efficient to find faster and better path for any transmission in future iterations. It will reduce energy consumption by the intermediate nodes since the distance between the source and the destination is minimized using localization by PSO.

The rest of the paper is organized as follows. We discuss related work in Section 2. In Sections 3, we have presented pre-requisites of the MECT-PSO algorithm. In Section 4, we present our proposed algorithm in brief. The performance of MECT-PSO protocol has been evaluated in Section 5. Section 6 finally includes concluding remarks over the performance of our system.

2. RELATED WORK

The efficiency of a routing protocol can be improved by controlling the topology of the network. Network topology control plays an important role in self-organizing wireless sensor networks [1]. A good organized network helps to save node energy and consequently prolong the lifetime of the network [2]. The main research problem is to control and position the nodes in an optimized way such that the network should satisfy the coverage and the connectivity issues under limited resources. Populated and scarce regions of the network should be made homogeneous so that unnecessary links between nodes are eliminated and every individual node in the network is reachable. The idea is to position nodes such that a single hop can reach as many as neighbours. Therefore, the routing strategy should be improved to select stable/reliable path to enhance the data transmission while conserving node energies for network survival in constrained environment. In most of the early stages of research, people paid attention on the initialization stage of topology rather than long running time of the network when using the topology control techniques. Due to random and unpredictable movement of the nodes, link quality is unstable and the nodes may fail eventually.

The computational complexities are function of problem size and thus degrades the performance of the traditional optimization techniques. The motivation to use computational intelligence algorithms comes into existence due to the surplus cost of programming engines and resource requirements for the resource constrained nodes. Various adaptive mechanisms for complex and dynamic environment of WSN have been proposed in the recent years for topology optimization such as evolutionary algorithms, swarm intelligence and neural networks. Some heuristic algorithms include GA (Genetic Algorithm, PSO (Particle Swarm Optimization, DE (Differential Evolution), BFA (Bacterial Foraging Algorithm), ACO (Ant colony Optimization), SOFM (Self Organizing Feature Maps), FS (Fish Swarm), WS (Wolf Swarm) *etc.* Nowadays, the work is concentrated in improving the resilience of network to faulty nodes and unreliable links without exhausting the limited resources [3].

S. R. Barkunan and V. Bhanumathi [4] proposed a cluster based efficient node deployment mechanism to improve network lifetime and other QoS parameters. They chose cluster based on number of neighbors and residual energy of the nodes. Omar Adil Mahdi *et al.* worked to find efficient paths from cluster head (CH) nodes to sink node on the basis of highest node performance ranking index to select the next hoping node [5]. The ranking index was calculated for each node depends on three parameters including residual energy, node depth and the void indicator (Number of neighbors). Intermediate nodes from source to receiver node are selected on the basis of trust values for legitimacy. Once a packet is transmitted, the trust value is updated from LSB to MSB in 8 bits trust vector. The trust value comprises of recommendation and the communication trust equivalent to the energy cost and cost for packet transmission, respectively. Reliability index used by Muhammad Ismail in [6] is a function of node energy, average energy in the next forwarding region, and shortest path index. The shortest path index comprises of number of hops to the sink and average depth of neighbours in next hop. The index helps to detect and limit the energy holes to mitigate the probability of packet loss and improves the network lifetime. Graph theory was proposed to find the shortest reliable path for link failure conditions due to intermediate node in low density networks by M. S. Nidhya and R. Chinnaiyan in [7]. Their work focused on finding shortest reliable alternative route to bypass the faulty node to prevent loss of packets. Suzan Shukry [8] worked to guarantee stability during data transmission by selecting stable forwarding nodes using betweenness centrality concept [9]. The forwarding node then acts as a source to activate another path routing mechanism to find another stable node using stable path routing metric until the packets are destined. Their approach was able to mitigate hot spots and outperformed other routing schemes such as local reliability-based routing and energy aware schemes. Rehan Almesaeed and Ahmed Jedidi proposed a dynamic directional routing to select the next hoping node based on area discovery phase and forwarding phase. A virtual pie shaped region is considered at each transmission attempt for finding the competent node to forward packets. The bounded pie region and active nodes then shares their information with the source regarding position, residual energy, received power level and availability. The next stage involves selection of proper candidate based on path loss factor, residual energy and received power level which ascertain that the same node will not be selected when next path discovery is initiated by the same source [10].

The work proposed in [11] used generalized tree topology for efficient routing using PSO. They concentrated on constructions of trees based on the distribution that the particles had in past topologies to evaluate the maximum connectivity of the particle built with Chow-Liu. Ying Zhang *et al.* in [12] improved static and dynamic network invulnerabilities against unsecured attacks using fireworks algorithm FW) having strong searching features and PSO with fast convergence. They combined the features of PSO and FW to generate new generations (positions) of the particles. The good performer particles from PSO are combined with particles of FW resulted as bad performers of PSO to form new swarm thus improving convergence speed and search ability. Performance comparison using PSO variants for topology optimization was presented in [13] for different population topologies. Discrete Binary version of Grey Wolves and Chicken Swarm optimization [14] were employed for minimizing the total number of active nodes and affirmation of sufficient residual energy respectively to maintain network coverage by Mohamed Mostafa Fouad *et al.* They were able to establish an efficient route with small numbers of active nodes prolonging the network lifetime. Mohamed Tounsi [15] selected active and energy efficient forwarding nodes based on Tabu search and minimum spanning tree routing algorithm. He considered the fact that few nodes deplete energy faster than others and choosing any node to forward packets considering residual energy is not sufficient. Nodes approaching the target are checked in the Tabu list (inactive nodes) for consideration to be selected for next hop. The list members are not selected unless they are omitted from the list. That is all other nodes covering the same target reach the same energy level. They showed that the network lifetime was increased by 45%. Osamah Ibrahim Khalaf *et al.* used Bee algorithm to obtain better coverage as compared to GA in some specific areas of the network [16]. The leading bees were forwarded to find solutions and stopped when the number of solutions were met with a predefined threshold of 50.

3. DESIGN OF MECT-PSO ALGORITHM

This section explicitly details the processing of the proposed MECT-PSO scheme. Specific design and statistical formulation is presented to refine the proposed scheme step by step. The notations and expressions included in the research work which are used in computation of various parameters are described below.

3.1 Preliminaries

3.1.1 The WSN network

SN – denotes total number of sensor nodes deployed in the network. $Ni \in \{N_0, N_1, N_2, N_3, \ldots, N_{SN\text{-}1}\}$ – denotes node ID's or node numbers. All the nodes are accessible considering geographical positioning system (GPS). Nodes exchange their information using control packets during initialization. An Intermediate Node is designated as IN whereas, IN's is used for Intermediate nodes along a selected path using route discovery mechanism. The nodes are randomly positioned at the instant of deployment.

3.1.2 MECT algorithm

The difference between MET and MCT algorithm [17] is that the former assigns tasks to resources or nodes without considering resource availability and the later assigns task randomly but are similar in nature to consider the best predictable completion time for that task. The MCT algorithm assigns tasks to nodes which have earliest completion time rather than minimum execution time. It overcomes the drawbacks of Opportunistic Load Balancing (OLB) [42] and the MET algorithm. On the other hand, the MET assigns task based on minimum execution time irrespective of the availability to the nodes which sometimes result in high load imbalance. The difference between the two is related with the words 'earliest/best' respectively. For MCT, it is earliest completion time while for MET, it takes on best completion time [17].

Let T is the Number of tasks to be completed such that $T_i \in \{T_0, T_1, T_2, T_3, \ldots, T_T\}$. R_n – represent ready time required for task T_m on node N_n, E_{mn} – denotes the execution time of task T_m on node N_n and C_{mn} – denotes the completion time of task T_m on node N_n. The relation between completion time, execution time and the ready time is expressed as:

$$C_{mn} = E_{mn} + R_n. \quad (1)$$

The arrival time of packet is simply the time when the packet reaches a sensor node for execution. Begin time represents the time when it is ready to be executed by the node. The Begin time is dependent on the status of Queue, execution time required to execute various packets by the sensor node and the availability of resources to complete various tasks over the network. We had considered heterogeneous nodes having different execution time for different tasks to perform. Therefore, the Begin time varies with every individual machine or node thus greatly affecting the routing mechanism. Finally, the completion time represents the total time required by the node to begin the execution and execute the task *i.e.*, sum of Begin time and the Execution time.

$C_{\min}$ – is the minimum completion time for any task T_p on a particular node N_n.
If P – denotes the number of possible paths from source to destination node, then for P possible paths,

$$P_i \in \{P_1, P_2, P_3, P_4, \ldots, P_P\}.$$

B_{IN} – begin time for an intermediate node along the path and it equals ($B_{IN} = R_n$). Q_T – time taken by any intermediate node to execute packets in its Queue and P_T – represents time taken to execute packets by any intermediate node delivered from the previous node, then

$$B_{IN} - Q_T + P_T. \quad (2)$$

$B_{pathnodes}$ – vector containing Begin time for all intermediate nodes along a path

$$B_{pathnodes} = \bigcup_{1}^{N_{IN}} B_{IN}. \quad (3)$$

Where, N_{IN} – number of intermediate nodes along an individual path. The maximum begin time required by an intermediate node along any single path is evaluated as,

$$B_{\max} = BP_{\max} = \max(BP_{pathnodes}). \quad (4)$$

BP_{path} – a vector holding maximum begin time for each path

$$B_{path} = \bigcup_{1}^{P} BP_{\max}. \tag{5}$$

$B_{\min}$ – is found from set BP_{path} which is equivalent to the least begin time required for any path from all possible paths P and is expressed as,

$$B_{\min} = \min_{P}(BP_{path}). \tag{6}$$

E_t – denotes the execution time for any path from P found from the task machine matrix and E_P – is a vector of all execution times corresponding to all possible paths P such that,

$$E_P \in \{E_{t1}, E_{t2}, E_{t3}, E_{t4}, \ldots, E_{tP}\}.$$

C_{tp} – denotes the completion time for any path from P and is calculated as,

$$C_{tp} = B_{IN} + E_t. \tag{7}$$

C_P – is a vector of all completion times corresponding to all possible paths P, such that

$$C_P \in \{C_{tp1}, C_{tp2}, C_{tp3}, C_{tp4}, \ldots, C_{tpP}\}.$$

The optimum path O_{path} is selected when minimum execution time is less than least begin time, otherwise it is selected based on minimum completion time and can be expressed as,

$$O_{path} = \begin{cases} P_i, & index\ i \in (E_{ti} < B_{\min}) \\ P_j, & index\ j \in (\min(C_P)) \end{cases}. \tag{8}$$

3.1.3 Particle swarm optimization

PSO was introduced by Eberhart and Kennedy [19] in 1995 based on swarm involving particles of a finite predefined numbers say N_p. Each D-dimensional particle $P_{n,d}$ in N_p ($1 \le n \le N_p$) is able to converge to solution to the multidimensional problem. Each particle $P_{n,d}$ has its position $X_{n,d}$ ($1 \le d \le D$) and velocity $V_{n,d}$ in the dth multidimensional space. Each of the individual particle is then subjected to a fitness function and the quality of the solution is judged with respect to a zero or permissible error. The particle corresponding to the best solution called the Global best (G_{best}) is monitored in each step while the particle personal or own best performance from all the previous steps is memorized called the Particle best (P_{best}). In order to reach the goal, each particle update their respective velocity $V_{n,d}$ and position $X_{n,d}$ using the current position (P_{curr}), global best position (G_{best}) and the particle own best position (P_{best}). The updated values of velocity and positions are then subjected to some lower and upper limits so that the particle remains in the search space and contributes in achieving the goal. The values of the limits depend on the minimization or the maximization optimization problem. The new values are then used to evaluate the fitness function for obtaining the new values of P_{best} *and the* G_{best}. The new position of a particle in the multidimensional search space is influenced by P_{best}, G_{best}, $X_{n,d}(t)$ *and* $V_{n,d}(t)$ as indicated in Fig. 2 below.

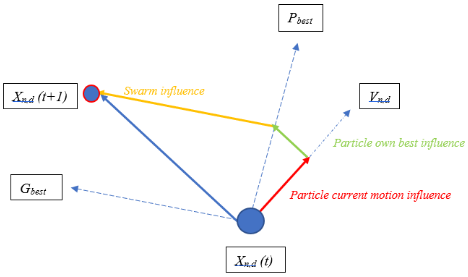


Fig. 2. Influence of various factors for particle new position.

$P_{n,d}$ – are particle D-dimensional space and the population is expressed as
$P_{n,D} = [X_{n,1}(t), X_{n,2}$(es in t), $X_{n,3}(t), X_{n,4}(t), X_{n,5}(t), \ldots\ldots\ldots X_{n,D}(t)]$
MaxIterations – Maximum iterations for PSO and k – to represent the current iteration. w – is the inertial weight factor (self-adapting parameter) and defined as,

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{MaxIterations} * k \tag{9}$$

where $w_{\max} = 0.9$ and $w_{\min} = 0.4$. c_1 and c_2 are acceleration constants and lie in the range $(0 \leq c_1, c_2 \leq 2)$, r_1 and r_2 are randomly generated values in [0 1].

The velocity $V_{n,d}$ and position $X_{n,d}$ in dimension D is updated using the following expressions,

$$V_{n,d} = w * V_{n,d}(k-1) + c_1 * r_1 * \{P_{best} - X_{n,d}(k-1)\} + c_2 * r_2 * \{G_{best} - X_{n,d}(k-1)\}, \tag{10}$$

$$X_{n,d}(k) = X_{n,d}(k-1) + V_{n,d}. \tag{11}$$

For minimization problem, the new value for P_{best} and G_{best} can be calculated as

$$P_{best} = \begin{cases} P_{curr}, & \text{if}\{fitness(P_{curr})\} < fitness(P_{best})\} \\ P_{best}, & \text{Otherwise} \end{cases}, \tag{12}$$

$$G_{best} = \begin{cases} P_{curr}, & \text{if}\{fitness(P_{curr})\} < fitness(G_{best})\} \\ G_{best}, & \text{Otherwise} \end{cases}. \tag{13}$$

3.1.4 Initializing particles

PSO is used to find an optimal position for the *IN*'s for the MECT optimized path so that the distance from source node to the destination node get minimized to save energy when actual packet transmission takes place. The positions of the nodes are governed by [X, Y] coordinates in the network space therefore each PSO particle is initialized with [X, Y] position with two element vectors in $D = 20$-dimensional space. That is, for each of the *IN* $SN = 20$ particles are considered to search for the optimum position. For instant, if number of *IN*'s are three $N = 3$, then the initial particles would require $3\times2\times20$ matrix where each row corresponds to a position for individual *IN*, column corresponds to the [X, Y] coordinates and the third dimension are for the number of particles for each of the node. The initial positions of the particles were set through a novel approach to find the goal (optimum position) as early as possible without exhausting the maximum iterations. Considering Fig. 3, the following points and distances are noted as:

$I(X, Y)$ – position of intermediate node I in the network space. D_d = 50m is the threshold distance that should be maintained between the intermediate node and its neighbor. $M(x_M, y_M)$ – is the midpoint on line between nth intermediate node and $(n+2)^{th}$ intermediate node. D_{rx} and D_{ry} are the direction indicator of midpoint M.
D_{IM} – is the distance between node $(n+1)^{th}$ intermediate node I and midpoint M. $(x_{\max}, y_{\max})$ – represents the maximum offset that node I can take w.r.t. its current position (X, Y). (x_{disp}, y_{disp}) – is the randomly generated displacement for node I to add w.r.t. its current coordinates (X, Y).
(X_n, Y_n) – New positioning coordinates for node I. D_{offset} – is the distance between current position at (X, Y) and new position at (X_n, Y_n). $D_{neigh_{old}}$ – is the vector of distances between $I(X, Y)$ and neighbors and $D_{neigh_{new}}$ – is the vector of distances between $I(X_n, Y_n)$ and corresponding neighbors. D_{PN} – represents distance between n^{th} intermediate node P and $(n+2)^{th}$ intermediate node N whereas D_{PIN} – represents distance between n^{th} intermediate node P to $(n+1)^{th}$ intermediate node I and $(n+1)^{th}$ intermediate node I to $(n+2)^{th}$ intermediate node N.

3.1.5 Fitness function for PSO

D_k is the distance between two successive nodes along the path. D_m – represents the distance travelled by an *IN* to acquire new position (X_n, Y_n) from its current position (X, Y). etr = 0.001 J/unit length, is the energy loss when a node travels from one position to another and etx = 0.01 J/unit length, is the energy loss when a node transmit packet. *AEL* – is the Average Energy Loss occurred due to the movement and transmission of packets by the intermediate nodes over a path and $AEL_{\min}$ – is the minimum value of *AEL* found from set of *AEL* values calculated for each possible path from source to destination.

4. PROPOSED WORK

MECT was recommended by [20] due to its low computational complexity. MECT helps to improve the scheduling algorithm performance by reducing the packet transmission time over the selected path from source to destination node. Algorithm 1 elaborates how an optimized path is selected from all possible paths obtained from source to the destination. Lines 30-34 in the algorithm selects the optimum path either based on minimum execution time or minimum completion time. The path with minimum overhead is selected to improve packet delivery and throughput with minimum amount of delay.

Algorithm 1: Optimum path selectin using MECT Algorithm

1: *Input* ← Q_T, P_T and Task Machine Matrix
2: *Output* ← B_{min}, E_P, C_P
3: Final Output ← Optimum path (Index)
4:
5: **for** *paths* = 1, 2, …, *P* **do**
6: **for** *pathnodes* = 1, 2, …, *IN* do

Compute Begin Time $B_{IN} \leftarrow Q_T + P_T$
Store $B_{pathnodes} \leftarrow B_{IN}$
end for
find Maximum Begin Time along the path
$BP_{\max} \leftarrow (B_{pathnodes})$
Store $BP_{path} \leftarrow BP_{\max}$
end for
find Minimum Begin Time from all paths
$BP_{\min} \leftarrow \min(BP_{path})$
This represents the least time required for any path from all paths to process the packet
for *paths* = 1, 2, …, P **do**
Compute path Execution time E_t using Task Machine Matrix
This excludes the Source and the Destination Node
Store $E_P \leftarrow E_t$
end for
for *paths* = 1, 2, …, P **do**
Compute path Completion time $C_{tpath} \leftarrow B_{IN} + E_t$
Store $C_P \leftarrow C_{tpath}$
end for
if $E_P < B_{\min}$ **then**
Select the path
else
Select the path with Minimum Completion Time
end if

4.1 Relocating Intermediate Nodes using PSO

Under dynamic networks, it is possible to direct the nodes to acquire new positions in the network space. The distance between the source and the destination node can be reduced to save energy during transmission of packets. The energy required to transmit a packet over a path depends on length of the path from sender to the receiver. The node relocating algorithm should find new position for the intermediate node in such a way that:

1. The path length is reduced considerably to reduce energy loss and delays.
2. It should not be positioned outside the transmission range of its sender node to prevent link failure.
3. It should not lose its own neighbors, but it can acquire new neighbors for better connectivity.
4. It should not acquire new position in close vicinity of any other node for better coverage.
5. Sum of its distance from its sender node and receiver node should be less than distance between sender and receiver node otherwise it would be extra overhead in terms of hop count.

Since the optimal path selected by MECT will consist variable number of *IN*'s, a solution to multidimensional constrained problem is required. The aim is to find optimum positions for all the *IN*'s under constraints 1 to 5 listed above. We had chosen PSO as an optimization tool since it offers better solution to constrained multidimensional problems. Fig. 3 below explains the mechanism.

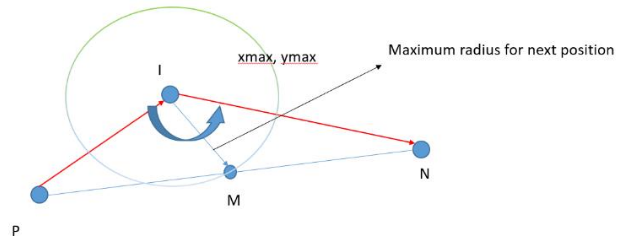


Fig. 3. Initial placement for particles.

Let *P*, *I* and *N* are the $(k–1)^{th}$, k^{th} and $(k+1)^{th}$ intermediate nodes along the energy efficient path obtained using MECT algorithm. *P* represents previous node/sender node and *N* represent the next node/receiver node with respect to node *I* for which the initial particle placement is required. The arrow (red color) indicates the direction of data from source *P* to the destination *N*. The question is to where intermediate node *I* should move so as to reduce the distance *P-I-N*. There may be infinite points around *I* in a circular range of radius (*TR* = 250m) defined by the transmission range. Therefore, to ensure early solution under constraints, node *I* was initialized in the region indicated by the angular arrow as shown in Fig. 3 and moved towards midpoint *M* on *PN* in every iteration. Algorithm 2 is used to obtain initial values of the particles (position of intermediate nodes). The constraints are implemented from lines 26-41 of the algorithm. The *IN* is displaced with respect to some random displacement. Failing to one of the constraint backs the algorithm to generate new random displacement. The position which satisfies all the constraints above is selected for the node *I* to take as the new position.

The best particle holds the best position $G_{best\text{-}position}$ for the intermediate node *I* corresponding to best fitness value $G_{best\text{-}value}$ calculated using AEL to be minimum. That was the initial position coordinates of node *I* and the energy loss at node *I*. We had considered the overhead due to travelling of node *I* from its current position to the new position. The PSO uses the fitness function (minimization) defined by the following expression:

$$AEL = \frac{1}{(N_{IN}-1)}\sum_{k=1}^{(N_{IN}-1)}(etx * Dk) + \frac{1}{(N_{IN}-2)}\sum_{m=1}^{(N_{IN}-2)}(etr * Dm) \quad (14)$$

Initially, $G_{best\text{-}position}$ = Coordinates of all the *IN*'s along the path (except Source and Destination Nodes).

$G_{best\text{-}value}$ = *AEL* found by *MECT* algorithm for each *IN*

The initial positions of the particles are initially the P_{best} values for all the particles before the PSO loop starts. After the PSO parameters have been initialized, the positions of all the *IN*'s are updated using the PSO algorithm in each step. The fitness function *AEL* corresponding to these new positions of *IN*'s is evaluated and compared with the $G_{best\text{-}value}$.

If the fitness *AEL* for any particle is less than $G_{best\text{-}value}$, $G_{best\text{-}position}$ and $G_{best\text{-}value}$ are updated. The algorithm run until all iterations are exhausted. The path index corresponding to $AEL_{min} = G_{best\text{-}value}$ is considered to be the optimized path. The velocities and the position of the particles are updated according to the following generalized expressions.

$$V_{new} = w * V_{old} + c_1 * r_1 * abs(P_{best\text{-}position} - X_{old}) + c_2 * r_2 * abs(G_{best\text{-}position} - X_{old}) \quad (15)$$

$$X_{new} = X_{old} + V_{new} \quad (16)$$

Where V_{old} – is the velocity acquired by particle in $(k-1)^{th}$ iteration and V_{new} – is current velocity that the particle should acquire, updated in k^{th} iteration. X_{old} – is the position acquired by particle in $(k-1)^{th}$ iteration and X_{new} – is current position that the particle should acquire, updated in k^{th} iteration.

Algorithm 2: Initialization of PSO Particle (Position of *IN*)

Reference – Fig. 4
Output – Initial position for intermediate nodes used by PSO

Assume the intermediate Node is at (X, Y)
Initialize distance $D_d \leftarrow 50$ for restricting close vicinity of two nodes
Calculate the distance from node P to node N
Get the midpoint between P and N as $M \leftarrow (x_M, y_M)$
Find the direction of this midpoint M w.r.t. node I using
$D_{rx} \leftarrow (X - x_M)$ and $D_{ry} \leftarrow (Y - y_M)$
Calculate Distance D_{IM} between node I and node M
This distance represents the radius of the circular region for the movement of node I
Compute maximum displacement the Intermediate node I can travel using
$x_{\max} \leftarrow abs(X - x_M)$ and $y_{\max} \leftarrow abs(Y - y_M)$
Randomly fine $(x_{disp}, y_{disp}) = randomize()$
$X_n \leftarrow (X + x_{disp})$ and $Y_n \leftarrow (Y + y_{disp})$

Calculate distance between (X, Y) and $(X_n, Y_n) \rightarrow D_{offset}$
if $D_{offset} > D_d$ **then**
    Go to generate new random displacement coordinate (x_{disp}, y_{disp})
end if

Compute Neighbor distances from $I(X, Y) \rightarrow D_{neigh_{old}}$
Compute Neighbor distances from $I(X_n, Y_n) \rightarrow D_{neigh_{new}}$

$flag \leftarrow 1$
if $D_{neigh_{old}} < D_d$ and $D_{neigh_{new}} < D_{neigh_{old}}$ **then**
    Discard the new position (X_n, Y_n), Generate (x_{disp}, y_{disp})
    $flag \leftarrow 0$
end if
if $D_{neigh_{old}} < D_d$ **then**
    Discard the new position (X_n, Y_n), Generate (x_{disp}, y_{disp})
    $flag \leftarrow 0$

33: **end if**
34: **if** $D_{neigh_{old}} < D_d$ and $D_{neigh_{new}} < D_{neigh_{old}} > TR$ **then**
35: Discard the new position (X_n, Y_n), Generate (x_{disp}, y_{disp})
36: $flag \leftarrow 0$
37: **end if**
38: **if** $D_{PN} < D_{PIN}$ **then**
39: Discard the new position (X_n, Y_n), Generate (x_{disp}, y_{disp})
40: $flag \leftarrow 0$
41: **end if**
42:
43: if $flag == 1$ then
44: Initialize Neighbor position at $I(X_n, Y_n)$
45: **end if**

$V_{\max}$ – is the maximum velocity that the particle can acquire, set to 20m/s while $V_{\min}$ – is the minimum velocity that the particle can acquire, set to 5m/s. Lines 20-26 represents the limits imposed on V_{new} and X_{new}.

The following PSO Algorithm 3 is used for finding the optimal positions for *IN*'s.

Algorithm 3: PSO based Optimal positioning of Intermediate Nodes (*IN*'s)

1: Input – Optimized path by MCET
2: Output – PSO Organized path
3:
4: Initialized PSO Parameters
5: MaxIterations ← 20
6: **for** $k = 1, 2, \ldots, MaxIterations$ **do**
7: New Path Distance $D_{new} \leftarrow$ Source to Destination
8: Avg Energy loss $AEL \leftarrow$ Transmission Loss + Travelling Loss
9: Find the particle with Minimum Average Energy loss $AEL_{\min}$
10:
11: **if** $AEL_{\min} < G_{best\text{-}value}$ **then**
12: Updated $G_{best\text{-}value}$ and $G_{best\text{-}position}$
13: Updated $P_{best\text{-}value}$ and $P_{best\text{-}position}$
14:
15: Calculate value of Intertrial Weight w
16: Update Velocities in V_{new}
17: Update Positions in X_{new}
18: **end if**
19:
20: **if** $V_{new} < V_{\min}$ or $V_{new} < V_{\max}$ **then**
21: $V_{new} \leftarrow randomized()$
22: end if
23:
24: **if** $X_{new} < X_{\min}$ or $X_{new} < X_{\max}$ **then**
25: $X_{new} \leftarrow randomize()$
26: **end if**

```
end for
Path selected with AEL_min
```

4.2 Description of the MECT-PSO Protocol

Based on the theories and equations discussed in earlier sections, the complete design steps for MECT-PSO algorithm for energy conservation and other Quality of service parameters (QoS) are presented in the following pseudo-code below. Our work is primarily focused on selecting reliable paths and optimizing topology. The complete design had been coded using MATLAB 2019b, over i5 Processor (1.8 Giga Hertz), 8 GB RAM and Windows 10 Environment.

Table 1. Network & node parameters.

Parameter	Value	Unit
Network size	1670 × 970	m^2
Number of sensor elements	50	Node
Initial energy of a node	100	Joules
Transmission range	250	m
Size of a packet	256	Bytes
Node velocity limits	[L-5 U-20]	m/s
MAC type	IEEE 802.11	–
Transmission Power	1.35	mW
Energy Loss due to travel of node	0.001	Joules/m
Energy Loss due to transmission of packet	0.01	Joules/m

1. Initialize Network Parameters: The following parameters listed in Table 1 are used for the WSN network. Number of nodes, size of packets and rounds are varied in accordance to various required output parameters for comparison.
2. Initialize the Task-Node matrix. It is assumed that the nodes are heterogeneous and requires different Execution time to execute packets. We have considered five different packets including RREQ, RREP, Data Packet, RRER (Link Failure) and Message type. The task machine matrix is composed of the time required to execute each of the packet by the nodes in the network. The Task-Node matrix is randomly initialized with different time for transmitting data packet in the range [500ms 1000ms]. For all other four packets the execution time is similar and initialized to 0.5, 0.5, 1e-3 and 1 second respectively for RREQ, RREP, RRER and HELLO.
3. Randomly deploy the nodes over the network space [1670 × 970]. Each node is associated with its node ID and spatial coordinates.
4. Compute the Average Energy (AE) of the network using the following expression

$$AE = \frac{1}{SN}\sum_{i=1}^{SN}(E_i). \tag{17}$$

Where, SN – are the total number of nodes, E_i – is the node Energy of i^{th} node. Initially, energy for all nodes are assumed to be 100J.

5. Identify the sensor nodes having energy less than *AE* and greater than or equal to *AE*. Participant nodes (P_n) having energy greater than or equal to *AE* are allowed for transmission, routing and reception of data, while the non-participant nodes (NP_n) are not allowed for any of the operation. This is to ensure that no node in the network die exhausting its energy before the others to maintain the overall energy of the network. Both the types of nodes (Participants marked by green and non-participants marked by magenta color) are depicted in Fig. 4.

$$Participant\ Nodes\ P_n = \bigcup_{i=1}^{SN} (Ei \geq AE) \tag{18}$$

$$Non\text{-}participantNodes\ NP_n = \bigcup_{i=1}^{SN} (Ei < AE) \tag{19}$$

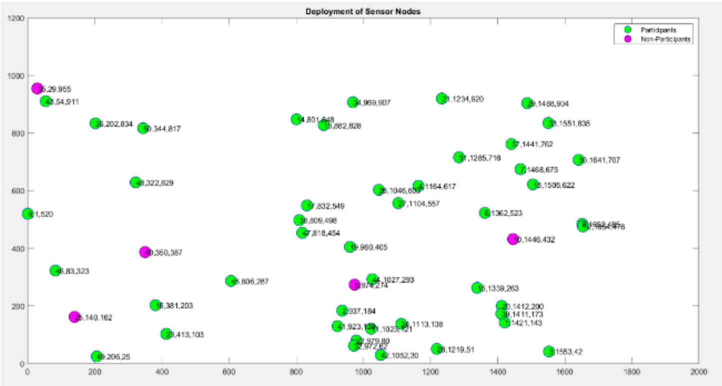


Fig. 4. Sufficient energy (Participants) and low energy (non-participants) nodes.

6. Identify the Source (*S*) and the destination (*D*) nodes for data transmission from the participant nodes. Form the request packet RREQ for route initialization. If *S* and *D* are same, display error message otherwise calculate the distance between *S* and *D* using the following distance equation.

$$D_s = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \tag{20}$$

where ($x1$, $y1$) are the coordinates of the *S* node, ($x2$, $y2$) are the coordinates of the *D* node and D_s is the distance between *S* and *D*. If the distance D_s is less than or equal to the transmission range (T_R), *S* can transmit directly to *D*, otherwise go to next step.
7. Calculate the *ADJACENCY* matrix. Here, we have removed the non-participant nodes from the adjacency matrix by first removing their neighbors and then eliminating them from the (50×50) matrix. Fig. 5 shows the source node (yellow), destination node (red), sufficient energy nodes (green) and the low energy nodes (magenta).
8. Find the shortest route from *S* to *D* with minimum hop count and also the alternative routes. To minimize the time complexity in finding alternative routes where some of the routes may contain large number of hops count and definitely may not be the optimal route, we adopted a novel way to find the alternative routes as mentioned below. Consider that the shortest route with minimum hops is: $S - N1 - N2 - N3 - D$ were $N1$, $N2$ and $N3$ are *IN*'s.

• We eliminated $N3$, keeping other IN's $N1$ and $N2$ along with S and D, updated the Adjacency matrix and found other possible paths. We eliminated $N2$, keeping other IN's $N1$ and $N3$, updated the Adjacency matrix and found other possible paths.
• Performing in the similar manner, we eliminated each IN at a time while keeping other in the list and found all possible paths from S to D. We have verified our results by finding all possible paths from S to D and comparing with results found through Dijkstra algorithm. It was found that the shorter path list was similar but our approach saved time.

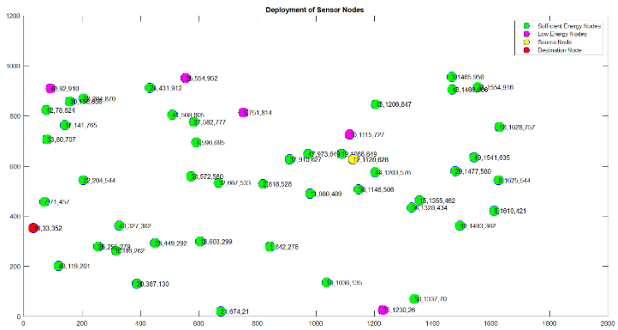


Fig. 5. Network with source, destination, participants and non-participants.

Possible path from Source node S (ID = 12) to Destination node D (ID = 16) found by the procedure mentioned above were:

Path 1 – 12 11 2 32 13 23 26 16 – Shortest path
Path 2 – 12 11 2 32 13 23 9 48 16 – Alternative path 1
Path 3 – 12 11 2 34 27 24 28 17 22 7 16 – Alternative path 2
Path 4 – 12 37 2 32 13 23 26 16 – Alternative path 3

It is seen that by eliminating IN's 2, 13 and 23 resulted in no path while eliminating 11, 32 and 26 resulted in three alternate paths form S to D.

9. Apply the MECT Algorithm for finding the optimal path out of all possible paths obtained in the previous step. Here the Queue is again randomly initialized with a buffer size of 10. The output of MECT algorithm is the optimal path as a result of Minimum Execution time or the Minimum Completion time. Table 2 shows various time elements calculated for all four paths:

Table 2. Begin time, execution time and completion time for all possible paths.

Path	Begin Time – BP_{path}	Execution Time – E_p	Completion Time – C_p
Path 1	**1834.0020**	4497.0000	7681.0090
Path 2	1834.0040	5251.0000	8559.5140
Path 3	2000.0020	6838.0000	14384.5140
Path 4	**1834.0020**	4274.0000	**6699.0090**

The minimum Begin time from column 2 is $B_{\min}$ = 1834.0020. None of the value of Execution time in column 3 is less than the minimum Begin time, therefore the MECT will look for a minimum value in last column 4, that is 6699.0090 which belongs to path 4. Hence, path 4 is selected according to Minimum Completion Time. Fig. 6 below shows the optimized path by MECT algorithm.

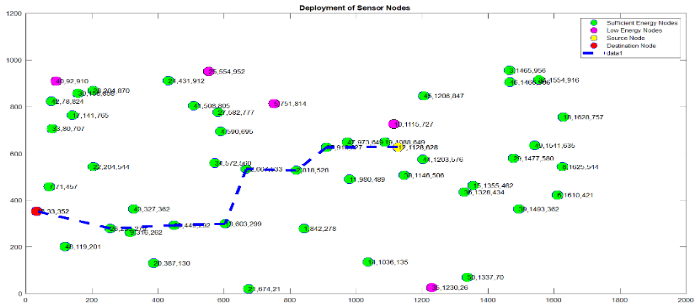


Fig. 6. Optimized path obtained by MECT algorithm.

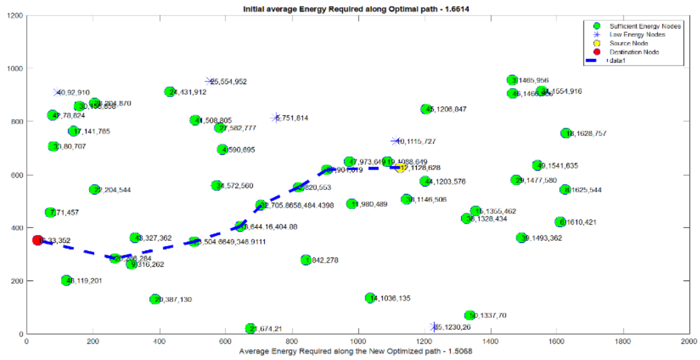


Fig. 7. Re-localization of *IN*'s using PSO optimization.

10. Apply Particle Swarm Optimization for localizing the *IN*'s of the MECT optimized path. Fig. 7 below shows the output of the PSO optimized path where the *IN*'s are relocated as compared to the original positions in Fig. 6 above. The initial energy (1.6614J) required to transmit a single packet along the MECT optimized path is reduced to 1.5068J with PSO optimization. By relocation, the alignment of the path has changed making it straighter while satisfying the constraints. The old and the new position of the *IN*'s are listed in Table 3.

Table 3. Initial node position and final position acquired by nodes through PSO in 20 iterations.

IN	Initial Position [*X*, *Y*]	New Position [*X*, *Y*]
37	(910.0000, 627.0000)	(904.0000, 619.0000)
2	(818.0000, 528.0000)	(820.0000, 553.0000)
32	(667.0000, 533.0000)	(705.8658, 484.4398)
13	(603.0000, 299.0000)	(644.1600, 404.8800)
23	(449.0000, 292.0000)	(504.6649, 346.9111)
26	(255.0000, 279.0000)	(266.0000, 284.0000)

Average Energy Loss along path selected by MECT – 1.6614
Average Energy Loss along path Optimized by PSO – 1.5068

11. Display the performance with respect to *AEL* and PSO iterations. Fig. 8 shows the initial *AEL* marked with red at 1.66 J and decrease in *AEL* with successive iterations.

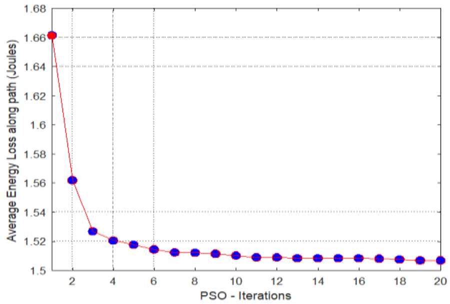
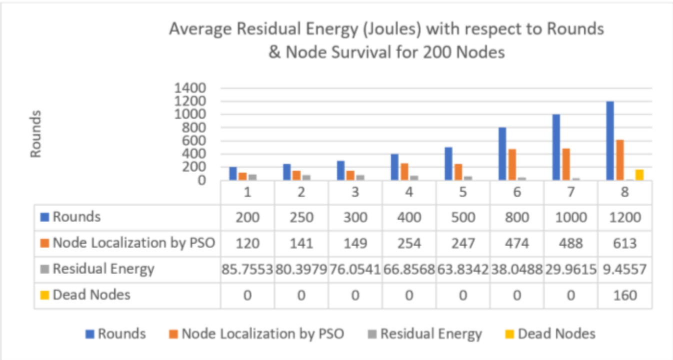


Fig. 8. Average energy loss due to node localization by PSO at each iteration.

5. PERFORMANCE EVALUATION

We simulated our system for 8 cycles of rounds (200, 250, 300, 400, 500, 800, 1000 and 1200) with 200 sensor nodes to evaluate the network performance. We restricted the simulation to 1200 rounds with the energy dissipation factors in joules ($etx = 10^{-2}$ J/m *and* $etc = 10^{-3}$ J/m) to reduce time complexity.



	1	2	3	4	5	6	7	8
Rounds	200	250	300	400	500	800	1000	1200
Node Localization by PSO	120	141	149	254	247	474	488	613
Residual Energy	85.7553	80.3979	76.0541	66.8568	63.8342	38.0488	29.9615	9.4557
Dead Nodes	0	0	0	0	0	0	0	160

Fig. 9. Performance of proposed MECT-PSO for 200 rounds with packet size of 256 bytes. Distribution shows actual rounds when path was optimized, the residual energy and number of dead nodes after each cycle of rounds.

We had considered a threshold of 10 Joules (Initial energy being 100 Joules) for a node to be assumed dead to reduce time and computational complexity. As seen from the statistics in Fig. 9, when the number of rounds are low at 200, 60% of total paths required optimization (120 out of 200) while when the number of rounds reached to 1200, the paths optimized were about 50% (613 out of 1200). It clearly indicates that the nodes find better position in the space around them to increase coverage and connectivity. The last column of the statistic clearly shows that 80% of nodes together are dead at the end of 1200th round keeping 20% surviving nodes.

5.1 Energy Saved under Different Node Densities.

To find average energy conservation as a function of node densities, we considered *SN*=100, 125, 150, 175 and 200 sensor nodes and set rounds to 25, 50, 75 and 100 individually. For scarcely populated network, the algorithm finds greater angular area and positioning length at any *IN*, whereas for densely populated networks, both the parameters are shortened and subjected to more neighborhood constraints. Average energy conserved is calculated for actual rounds which considers energy conserved with and without optimization. The path selection process in our approach guarantees for the minimum execution or the minimum completion time path but relocation of *IN*'s using optimization are subjected to constraints and are not guaranteed for every path. The average energy conserved for 100 nodes is higher while it is least for 200 nodes shown in Fig. 10. It means that MECT-PSO will save more energy for low density network as compared to high density networks. For densely populated network, MECT-PSO have to perform in strict constrained environment and the nodes have low scope of repositioning themselves.

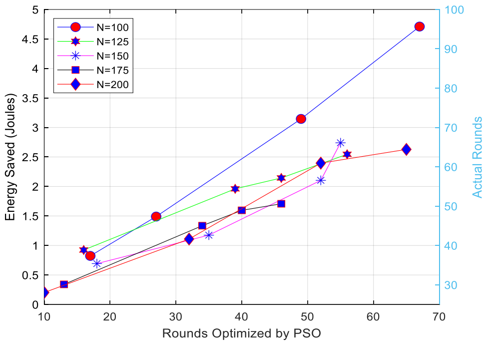


Fig. 10. Performance of proposed MECT-PSO for variable rounds (packet size of 256 bytes) and energy saved with optimization for different sensor node densities.

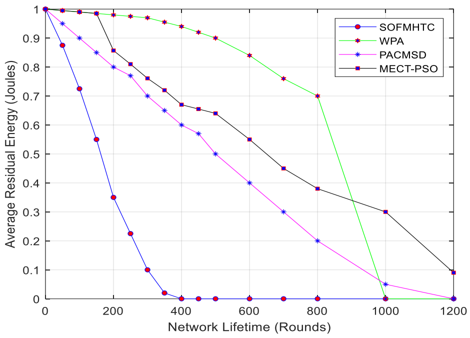


Fig. 11. Average residual energy as a function of rounds.

5.2 Average Residual Energy of Nodes

It is an important indicator to evaluate whether the self-organizing mechanism is superior. An efficient topology control mechanism can improve the remaining or residual energy of the sensor node. The simulation results of the average residual energy of proposed work and three other state of art work in SOFMHTC [21], WPA [22] and PACMSD [23] are shown in Fig. 11. The average residual energy of MECT-PSO and WPA is higher than that of the SOFMHTC and PACMSD algorithms. The WPA algorithm works fine till 800 rounds and drastically falls to zero at 1000 rounds while the network is alive with MECT-PSO at the end of 1200 rounds. The fact to be considered is the energy dissipation factors. The parameters *etr* and *etx* considered are of the order 10^{-9} as compared to our values of the range 10^{-3}. For better comparison, when we considered initial energy for the sensor node to be 1 Joule with *etr* and *etx* of the order 10^{-9}, the residual energy lasted up to 60000 rounds.

5.3 Number of Alive/Active/Surviving Nodes

The network performance depends on surviving nodes. More surviving nodes with time lengthen the life of the network. WSN nodes have the characteristic of random deployment and wide range. The energy consumption increases with time and sensor nodes tends to fail. It becomes necessary to increase the lifetime of the network nodes. Fig. 12 shows the comparison with respect to number of active nodes in the four algorithms. It is clear that the number of active nodes in the network decreases with time. After 200 rounds, all the algorithms do not lose any node. The performance of MECT-PSO and PACMSD is similar till 800 rounds and all the deployed nodes are alive. The number of survival nodes at 800th round for WPA is 140 while at 1000th round the surviving nodes PACMSD retains 120 active nodes. The MECT-PSO proposed in this article still maintains 200 nodes at 1000th round and falls at 1200th round to 40. Therefore, from the comparison of the surviving nodes with three other algorithms, it is clear that the topology optimization scheme proposed in this article has the longest network survival or lifetime. It is also clear that there is no gradual decrease of active nodes during successive rounds. The network will fail at 1200th round leaving only 20% of the nodes.

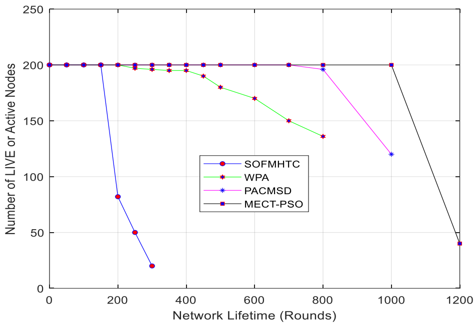


Fig. 12. Surviving nodes as a function of rounds.

5.4 Network Coverage

The quality of topology depends on network coverage. Many standard deployment schemes have been listed in the literature to provide better coverage and connectivity. But with the progress of time, the sensor nodes lose energy and falls outside the sensing range and the coverage is minimized. WPA algorithm in [22] outperforms PSO and AFSA algorithms by finding the optimal solutions in less iterations. We compared our proposed MECT-PSO algorithm with WPA as shown in Fig. 13 and found that the average coverage of the network has boosted by approximately 6%. On the other hand, MECT-PSO requires few iterations to find optimal positions for the intermediate nodes. Also, the response time is shortened and the routes are searched and optimized rapidly with improved stability. Osmah Ibrahim Khalaf *et al.* in [16] proposed BEE algorithm to improve the coverage problem and obtained the coverage percentage of 94.96% as compared to 89.63% with Genetic algorithm with 32 sensor nodes in 51.78 seconds. From Fig. 13, we obtained the coverage percentage of 98.12% in 1000 iterations which consumed approximately 13.69 seconds only.

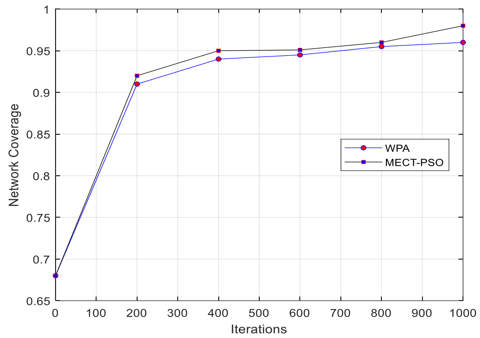


Fig. 13. Network coverage and convergence.

6. CONCLUSION

The proposed MECT-PSO algorithm aims to improve energy conservation, congestion control, stable links and network lifetime. Our approach had adopted the participant property where a node with energy higher than the average energy of the network participate in data transmission, reception and routing. We relaxed the minimum hop criteria for selecting the path between two communicating nodes and chose the path with minimum overheads. This improved the link life time and finds stable and reliable paths. Our approach restricts a node to participate frequently and prevent it from exhausting. On the other hand, the algorithm finds optimum positions in the network area to position the *IN*'s along the selected route such that the path length is reduced thus decreasing the amount of energy required to send data which in turns increases the network lifespan. Randomly positioned nodes get well organized as the iterations progresses and results in an organized form of network for better coverage and connectivity.

REFERENCES

1. P. K. Sahoo, S. R. Pattanaik, and S. L. Wu, "Design and analysis of a low latency deterministic network MAC for wireless sensor networks," *Sensors*, Vol. 17, 2017, pp. 2185.
2. J. Yan, X. Yang, X. Luo, and C. Chen, "Energy-efficient data collection over AUV-assisted underwater acoustic sensor network," *IEEE Systems Journal*, Vol. 12, 2018, pp. 3519-3530.
3. H. Zhao and L. Guo, "Energy saving of wireless sensor network based on topology control algorithm," *IEEE Access*, Vol. 7, 2019, pp. 85525-85535.
4. S. R. Barkunan and V. Bhanumathi, "An efficient deployment of sensor nodes in wireless sensor networks for agricultural field," *Journal of Information Science and Engineering*, Vol. 34, 2018, pp. 903-918.
5. O. A. Mahdi, A. B. Ghazi, and Y. R. B. Al-Mayouf, "Void-hole aware and reliable data forwarding strategy for underwater wireless sensor networks," *Journal of Intelligent Systems*, Vol. 30, 2012, pp. 564-577.
6. M. Ismail *et al*., "Reliable path selection and opportunistic routing protocol for under

water wireless sensor networks,” *IEEE Access*, Vol. 4, 2016, pp. 100346-100364.
7. M. S. Nidhya and R. Chinnaiyan, “Shortest reliable path for wireless sensor network,” *International Journal of Pure and Applied Mathematics*, Vol. 117, 2017, pp. 105-109.
8. S. Shukry, “Stable routing and energy-conserved data transmission over wireless sensor networks,” *EURASIP Journal on Wireless Communications and Networking*, Vol. 36, 2021.
9. L.C. Freeman, “Centrality in social networks conceptual clarification,” *Social Networks*, Vol. 1, 1978, pp. 215-239.
10. R. Almesaeed and A. Jedidi, “Dynamic directional routing for mobile wireless sensor networks,” *AdHoc Network*s, Vol. 110, 2021, p. 102301.
11. M. S. M. Hasnawi and A. A. Ibrahim, “Efficient routing technique in wireless sensor network using PSO topology optimization,” in *Proceedings of the 4th International Symposium on Multidisciplinary Studies and Innovative Technologies*, Vol. 2, 2020, pp. 1-4.
12. Y. Zhang *et al*., “FW-PSO algorithm to enhance the invulnerability of industrial wireless sensor networks topology,” *Sensors*, Vol. 20, 2020, p. 1114.
13. C. Cao, Q. Ni, and X. Yin, “Comparison of particle swarm optimization algorithms in wireless sensor network node localization,” in *Proceedings of International Conference on Systems, Man, and Cybernetics*, 2014, pp. 252-257.
14. M. M. Fouada, A. I. Hafez, and A. E. Hassanien, “Optimizing topologies in wireless sensor networks: A comparative analysis between the grey wolves and the chicken swarm optimization algorithms,” *Computer Networks*, Vol. 163, 2019, p. 106882.
15. M. Tounsi, “A dynamic heuristic for WSNs routing,” *Cogent Engineering*, Vol. 8, 2021, p. 1919040.
16. O. I. Khalaf *et al*., “Optimization of wireless sensor network coverage using the bee algorithm,” *Journal of Information Science and Engineering*, Vol. 36, 2020, pp. 377-386.
17. S. H. H. Madni, M. S. Abd Latiff, M. Abdullahi, S. M. Abdulhamid, M. J. Usman, “Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment,” *PLoS ONE*, Vol. 12, 2017, p. e0176321.
18. G. Patel, R. Mehta, and U. Bhoi, “Enhanced load balanced min-min algorithm for static meta task scheduling in cloud computing,” *Procedia Computer Science*, Vol. 57, 2015, pp. 545-553.
19. R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, 1995, pp. 39-43.
20. H. D. Kim and J. S. Kim, “An online scheduling algorithm for grid computing systems,” *Grid and Cooperative Computing*, LNCS, Vol. 3033, 2004, pp. 34-39.
21. Z. Chen, S. Li, and W. Yue, “SOFM neural network based hierarchical topology control for wireless sensor networks,” *Journal of Sensors*, Vol. 2014, 2014, p. 121278.
22. S. Wang, H. You, Y. Yue, and L. Cao, “A novel topology optimization of coverage-oriented strategy for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, Vol. 17, 2021, No. 1550147721992298.
23. S. Lu, M. Wu, and M. Zhao, “Particle swarm optimization and artificial bee colony algorithm for clustering and mobile based software-defined wireless sensor networks,” *Wireless Networks*, Vol. 28, 2022, pp. 1671-1688.

Poonam T. Agarkar had completed her Master in Technology in VLSI Design and currently pursuing Ph.D. under Electronics Engineering at Yeshwantrao Chavan College of Engineering, Maharashtra, India. Her research interests include wireless sensor networks and communication networks.

Manish D. Chawhan had completed his Ph.D. and published research papers in conferences and journals. He is an Associate Professor in Electronics and Telecommunication Engineering and working on granted research project on Wireless Sensor Network at Yeshwantrao Chavan College of Engineering, Nagpur, Maharahtra, India. He is recognized Ph.D. Supervisor with Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur, Maharashtra, India.

Kishor D. Kulat is a Professor with the Department of Electronics and Communication Engineering, Visvesvaraya National Institute of Technology, Nagpur, Maharashtra, India. He has authored or co-authored more than 100 papers in international/national journals and conferences. His current research interests include wireless, communication systems, networking, and real-time embedded system design. He had guided many research scholars.

Pratik R. Hajare had completed his Ph.D. in the area of Artificial Intelligence and published research papers in conferences and journals in various fields. He had more than 25 years of teaching and administration experience with Department of Electronics and Telecommunication Engineering. He is a supervising Ph.D. scholar at Mansarovar Global University, Bhopal, M.P., India and worked on many research projects.