Received 23 May 2019
Revised 8 July 2019
Accepted 11 July 2019

Energy-efficient genetic algorithm variants of PEGASIS for 3D Wireless Sensor Networks

Aaqil Somauroo and Vandana Bassoo
Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Mauritius, Reduit, Mauritius

Abstract

Due to its boundless potential applications, Wireless Sensor Networks have been subject to much research in the last two decades. WSNs are often deployed in remote environments making replacement of batteries not feasible. Low energy consumption being of prime requisite led to the development of energy-efficient routing protocols. The proposed routing algorithms seek to prolong the lifetime of sensor nodes in the relatively unexplored area of 3D WSNs. The schemes use chain-based routing technique PEGASIS as basis and employ genetic algorithm to build the chain instead of the greedy algorithm. Proposed schemes will incorporate an energy and distance aware CH selection technique to improve load balancing. Clustering of the network is also implemented to reduce number of nodes in a chain and hence reduce delay. Simulation of our proposed protocols is carried out for homogeneous networks considering separately cases for a static base-station inside and outside the network. Results indicate considerable improvement in lifetime over PEGASIS of 817% and 420% for base station inside and outside the network respectively. Residual energy and delay performance are also considered.

Paper type Original Article

1. Introduction

Breakthroughs in the field of microelectromechanical systems have led to the conception of cheap and autonomous devices with sensing capabilities. These devices which are very small in nature, are called sensor nodes. Nodes carry out collection of data from the environment through sensing and monitoring. As transmission of data to the base station is a collaborative effort, nodes form a network referred to as Wireless Sensor Network (WSN). Being increasingly cost and size effective, WSNs have seen their application expand to include vastly different areas ranging from enemy detection military applications to heart rate monitoring in healthcare.

Declaration of Competing Interest: None.

Publisher note: The publisher wishes to inform readers that the article “Energy-efficient genetic algorithm variants of PEGASIS for 3D Wireless Sensor Networks” was originally published by the previous publisher of *Applied Computing and Informatics* and the pagination of this article has been subsequently changed. There has been no change to the content of the article. This change was necessary for the journal to transition from the previous publisher to the new one. The publisher sincerely apologises for any inconvenience caused. To access and cite this article, please use Somauroo, A., Bassoo, V. (2019), “Energy-efficient genetic algorithm variants of PEGASIS for 3D Wireless Sensor Networks”, *Applied Computing and Informatics*. Vol. ahead-of-print No. ahead-of-print. https://10.1016/j.aci.2019.07.002. The original publication date for this paper was 13/07/2019.

Applied Computing and Informatics
Vol. 19 No. 3/4, 2023
pp. 186-208
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI 10.1016/j.aci.2019.07.002

The block diagram of a sensor node is shown in Figure 1. Sensors are of various types and measure some physical quantity such as temperature, motion, pressure and humidity. Memory is needed to store intermediate sensor readings and also packets from other nodes. The controller is tasked with collecting data from the sensors and processing it. It decides when and where by using routing protocols. It is also responsible for data aggregation. Batteries provide the power source in a node and since WSNs are often deployed in remote areas, they cannot be usually replaced or recharged. The transceiver is responsible for the transmission and reception of data such that nodes can form a network.

As mentioned previously, batteries cannot be replaced making sensor nodes energy-constrained devices. Hence, being as energy efficient as possible is of prime importance. To this end, most research focus on energy efficient MAC and routing protocols. This paper will be concerned with the development of power-aware routing protocols by adopting strategies like clustering, data aggregation, multi-hop transmission to optimise network lifetime and energy efficiency in WSN routing protocols.

One of the approaches in solving optimisation problems is the use of bio-inspired algorithms that are modelled on processes and behaviours which occur in nature. These algorithms are numerous and solve a wide variety of problems in WSNs [2]. This paper will make use of genetic algorithm.

The proposed routing protocols bring the following contributions:

1. Clustering is implemented to reduce long chains and thus improve delay performance.
2. A genetic algorithm is used to build a near minimal length chain for transmission of data.
3. A new cluster head (CH) selection mechanism is devised to improve load balancing in diverse network types.
4. The extension of PEGASIS and variants to three-dimensions.

The remaining of the paper is organised as follows. Section 2 reviews literature on WSNs and describes various existing routing protocols. Section 3 presents the system model and proposed approach. Section 4 shows and discusses the obtained results. Section 5 concludes with an overall evaluation of the work and presents future research opportunities.

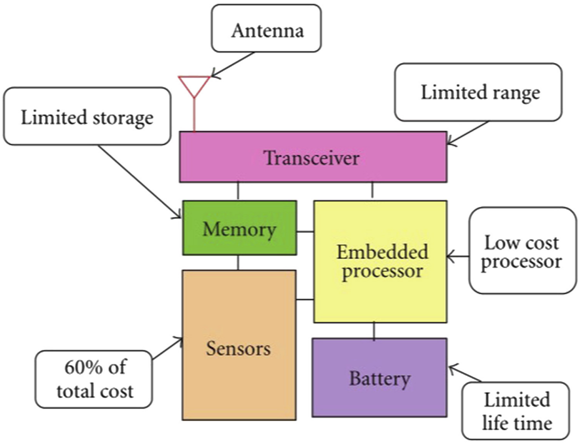

Figure 1. Wireless sensor node block diagram [1].

2. Related works

2.1 PEGASIS

Lindsey and Raghavendra [3] suggest using a chain-based protocol called PEGASIS. The chain is built such that all nodes are joined together using the greedy algorithm. This process of chain building is essentially known as the Travelling Salesman Problem (TSP). It is assumed that global knowledge of the whole network is available to all nodes. The chain building is started at the sensor node located furthest from the BS. The greedy algorithm entails finding a local minima at each step in a bid to eventually achieve a global minima. This means that the start node will form the first segment of the chain by joining to the closest node. This process repeats itself until finally reaching the end node. Chain reconstruction is carried out in the same way at the death of a node.

A leader is selected randomly from the one of the nodes in the chain. The leader node is responsible for transmission to the BS. It is important to note that the leader is rotated every round in a round robin manner. To clearly explain this concept, a list of all nodes may be considered. Once a node is selected as leader, it is crossed out on the list and it can only be selected as leader again when a new list is used. This occurs only when all nodes have been crossed out in the current list. In PEGASIS, data aggregation is carried out at every node except for the nodes located at the two ends of the chain.

The transmission mechanism in PEGASIS is triggered by the use of a control token issued by the leader and is passed on along the chain. To illustrate this, consider Figure 2 where C_2 is the leader node. The token is passed to C_2 and then to C_0. Node C_0 will then return back the token along with data to C_1. C_1 will in turn send data and the token to C_2. After reception of this data, C_2 will pass the token to C_4 along the chain and the same process which occurs at the other end of the chain takes place here too. It is important to reiterate that data aggregation occurs at every node except nodes at the end of the chain.

2.2 PEGASIS variants

PEGASIS is a classical protocol from which numerous variants have been developed to minimise its limitations.

2.2.1 PEG-ant. To build the chain, the greedy algorithm is used in PEGASIS. However, this chain is not optimal as it proceeds in a way that achieves local rather than global minimisation. This process of chain building is essentially solving the Travelling Salesman Problem (TSP). Guo et al. [5] propose PEG-Ant which uses ACO to build a smaller overall chain to decease transmission distance and hence save energy.

ACO is a nature-inspired algorithm from ants. The latter communicate with each other by releasing a substance called pheromone. The results highlight the clear superiority of using ACO to build the chain over the greedy algorithm, showing a marked improvement in the lifetime of the network. Other research have also made use of ACO to build the chain in PEGASIS [6,7].

2.2.2 IEEPB. While the CH rotation mechanism in PEGASIS is helpful towards load balancing, it does have some limitations. The nodes which are very far from the BS will deplete their energy faster than the nodes closer to the BS even though they have been chosen as the leader node the same number of times. To remedy to this problem, Feng et al. [8] used a

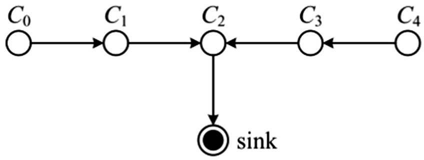

Figure 2.
Token passing scheme [4].

method where both the residual energy of a node and its distance from the BS is considered. A weighted score is obtained by the summation of a weighted energy and distance value. They also state that a flexible weight for the energy and distance enables the application of this method of leader selection to a wide variety of network situations with different requirements.

2.2.3 CRBCC. The long chain in PEGASIS causes large delays. Zheng and Hu [9] propose CRBCC which makes use of simulated annealing to build the chain. It also incorporates a clustering mechanism which aims to break the single long chain into multiple ones thereby reducing delay. Additionally, multihop inter-cluster communication is adopted as illustrated in Figure 3. The results show a reduction in the energy consumption of CRBCC in contrast to PEGASIS. It is also imperative to highlight the large improvement in delay performance for networks of different sizes for CRBCC.

2.2.4 PEG-ACO vertical clustering. The PEGASIS variants discussed previously aim to mitigate the limitations of PEGASIS. They address one, or at most two, of our stated limitations. Ramluckun and Bassoo [10] propose a very elegant solution incorporating some concepts from all three previously discussed protocols.

PEG-ACO Vertical Clustering makes use of clustering according to the x-coordinates of nodes. A chain is built in every cluster by using ACO as shown in Figure 4. Multihop transmission is adopted by CHs to send the data to the BS. All CH are chosen according to a weighted score between energy and distance. The performance of the protocol is assessed and improvements are obtained in lifetime, residual energy and delay. Another type of clustering method, Horizontal Clustering which always elects the CH responsible for transmission to BS in the uppermost cluster is also proposed. It is to be noted that the analysis is carried out solely for homogeneous networks with a static BS situated outside the network. While not always possible due to the nature of the application, having a BS inside the network increases its lifetime. Hence this work will consider both cases separately.

2.3 3D WSN

Most routing protocols found in WSN literature consider routing protocols only in two dimensions albeit the fact that we live in a 3D world. Roy and Mukherjee [11] remark that 3D rather than 2D design is required for many applications. One example is the design of underwater WSNs [12]. Other possible applications include deploying WSNs on trees with varying heights, railway tunnels and underground mines [11]. WSNs deployed in uneven

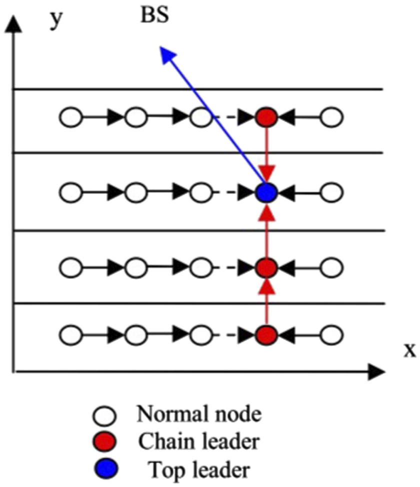

Figure 3.
Clustering into multiple chains [9].

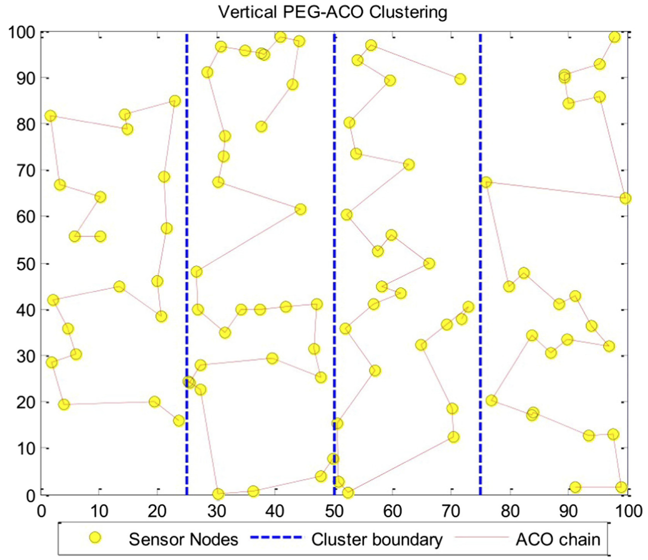

Figure 4.
Vertical Clustering and ACO chains [10].

terrains also fall under that category. Literature focuses on the coverage problem in 3D and research on routing protocols is relatively sparse. Kohli et al. [13] and Baghouri et al. [14] both propose extending LEACH to 3D. Hence, extension of PEGASIS to 3D would be a worthwhile research opportunity to enhance energy efficiency in 3D applications. To the best of our knowledge, this has not been carried out before.

2.4 Genetic algorithm

To solve the TSP in PEGASIS, a genetic algorithm will be used such that a near optimal chain length is obtained. Genetic algorithm is easy to implement and also less computationally expensive compared to techniques such as ACO [15]. It is inspired from nature and it attempts to imitate the processes which take place in Darwinian evolution. A chromosome is used to represent the solution and a population which contains multiple chromosomes is used. A chromosome contains genes and in the context of TSP, a gene is an index representing a specific node.

A solution is the list of nodes in a specific order. Adjacent nodes in the list are towns which must be visited one after the other or, in the context of PEGASIS, nodes which are joined together in the chain. The goodness of a solution is determined by a fitness function. In the present case, it is simply the distance covered when touring the nodes in the order specified by the chromosome.

In this paper, the GA will be applied to solve TSP in three dimensions. Also in the case of protocols which use clustering, the GA will be applied to each cluster separately to find the shortest chain length. In so doing, the overall chain length will be minimised aiding to improve energy efficient in 3D WSNs.

3. Methodology

3.1 Network model

N sensor nodes are randomly dispersed (non-deterministic deployment) in a square field whose length is M as shown in Figure 5 and with a static base station. Nodes are considered to

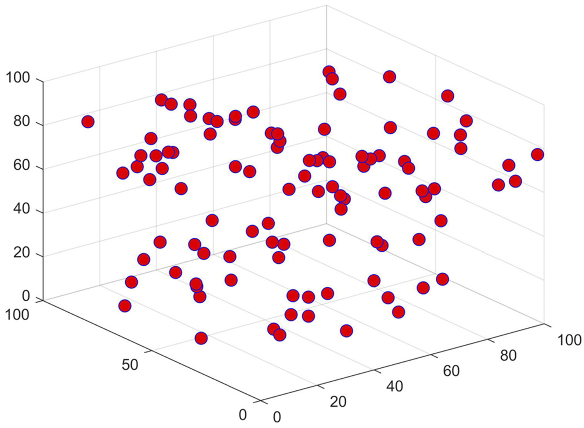

Figure 5.
Network model of 100 nodes in a 100 m × 100 m × 100 m 3D space.

be static once deployed. All nodes are considered to have a fixed amount of energy E_o. All parameters are specified in Section 3.7.

3.2 Radio and energy model

The first-order radio model shown in Figure 6 is used to calculate energy consumption of communicating nodes. It is simple to implement and widely used in WSN literature [16]. Hence, comparisons to previous protocols are made easier and fairer.

Transmitter and receiver circuits need energy to be run. The latter is a function of the number of bits (k) only [18].

$$E_{Tx/Rx-elec} = E_{elec} \times k \tag{1}$$

where E_{elec} is the energy expended per bit to run the transmitter or receiver.

The transmission energy used by the amplifiers is a function of both the number of bits and transmission distance (d) [18].

$$E_{Tx-amp} = \begin{cases} \epsilon_{fs} \times k \times d^2, & \text{if } d < d_0 \\ \epsilon_{mp} \times k \times d^4, & \text{if } d > d_0 \end{cases} \tag{2}$$

d_0 is a threshold used to determine which fading model is used (free space or multipath) and is given by:

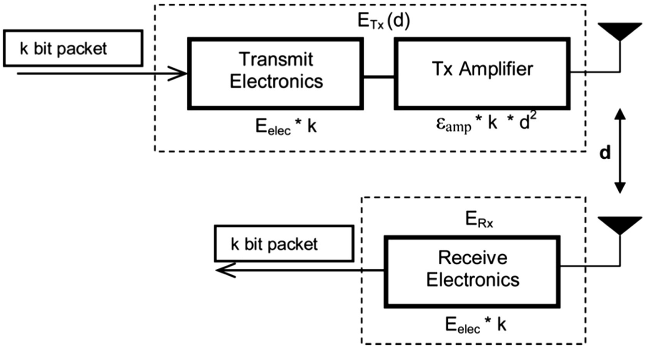

Figure 6.
First-order radio model [17].

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{3}$$

where ϵ_{fs} and ϵ_{mp} are constants which are used in the free-space and multipath model respectively.

Hence, the total energy required for transmission is:

$$E_{Tx} = \begin{cases} E_{elec} \times k + \epsilon_{fs} \times k \times d^2, & \text{if } d < d_0 \\ E_{elec} \times k + \epsilon_{mp} \times k \times d^4, & \text{if } d > d_0 \end{cases} \tag{4}$$

The energy required for reception consists of energy required to run the circuitry only.

$$E_{Rx} = E_{elec} \times L \tag{5}$$

The energy cost for data aggregation is [19]

$$E_{DA-tot} = s \times k \times E_{da} \tag{6}$$

where E_{da} is the energy per bit expended in aggregating and s the number of signals being aggregated.

3.3 Genetic algorithm

The GA is an iterative process. To find new solutions which may yield a better solution, GA operators are applied on chromosomes at each iteration. The simplest GA operators are [20]:

1. Selection This involves favouring the selection of solutions having better fitness.
2. Crossover This is similar to the reproduction process which takes place in evolution. It involves taking more than one solution and combining them to produce a new solution.
3. Mutation This entails modifying a portion of the solution and it helps to ensure diversity in a population [21]. It consist of the following:
   (a) Swap operation Two nodes are selected at random in the solution and their positions are swapped. For instance, considering positions 2 and 5

   a → b → c → d → e → f becomes a → e → c → d → b → f

   (b) Flip operation Part of the solution is flipped randomly as illustrated below:

   a → b → c → d → e → f becomes a → e → d → c → b → f after flipping at positions 2 and 5

   (c) Slide operation Part of the solution is selected randomly and it is ‘slid’ across to modify the solution. For illustration, the part of the solution from position 2 to 5 is considered.

   a → b → c → d → e → f becomes a → c → d → e → b → f

An open source code was used to implement the GA for chain building Kirk [22]. It makes use of selection and mutation. The steps in the GA algorithm are outlined below.

1. Initialise the initial population.
   The initial population contains multiple solutions which are generated in a random manner. Variable globalMin indicates the shortest chain length that has been achieved so far and it is assigned to infinity.
2. Calculate the fitness of each member of the population.
   populationMin is assigned to infinity. populationMin is a variable which indicates the best fitness value (smallest overall chain length) in the current population.
3. Record solution having best fitness in bestPopSolution.
   bestPopSolution is a variable which records the best solution in the current population. populationMin is assigned the value of best fitness.
4. If populationMin < globalMin, a new value globalMin is recorded as populationMin and the globalBestSolution is recorded as bestPopSolution. Variable globalBestSolution indicates the best solution which has been achieved so far.
5. The order of solutions in the population are randomised.
   For instance, solution 1 can become solution 30 and so on.
6. For every four members of the population(solutions 1–4, 5–8 and so on), the best solution is taken and the following steps are taken to create 4 new routes
   (a) Create random route insertion points. These consist of 2 points which indicate where a solution will be modified.
   (b) The best solution of the initial 4 is taken as new route 1.
   (c) Flip operation is applied to the best solution to create new route 2.
   (d) Swap operation is applied to the best solution to create new route 3.
   (e) Slide operation is applied to the best solution to create new route 4.
7. Hence, a new population is formed consisting of new solutions.
8. Steps 2 to 7 are repeated until the total number of iterations are completed.

3.4 Proposed CH selection

PEGASIS chooses its cluster head based upon a rotation policy. Whilst this somewhat aids in load balancing, it still proves to be somewhat inefficient. Hence, numerous other methods have been devised for choosing a more suitable CH. One method proposed by Ramluckun and Bassoo [10] computes a weighted score from the distance to base station and residual energy of the node which are both given weights. In the case of protocols with inter-cluster transmission, the distance from nearest CH may be considered if the cluster is a secondary cluster. Primary CH is hereby defined as the CH which transmits to the BS and the primary cluster as one which contains the primary CH. Other CHs which do not transmit to the BS are defined as secondary CHs. For ease of reference, the term CH for PEGASIS will be used instead of Cluster Leader.

Ramluckun and Bassoo [10] use the same weight to select both a primary and secondary CH. Additionally, the weights assigned were always equal to 0.5 to give residual energy and distance an equal say in the CH selection process. Intuitively, it would appear that having equal weights (0.5) for both energy and distance will yield best results. However, this is not necessarily the case and hence a different approach will be used here. It entails using assigning different weights to determine primary CH and secondary CHs. Also, the weights will not always be equal to 0.5.

Score for primary cluster-head

$$\text{Score}(\text{primary}) = w_1 \times E_{res}(\text{primary CH candidate}) + \frac{w_2}{d_{BS}} \tag{7}$$

Score for secondary cluster-head

$$\text{Score}(\text{secondary}) = w_3 \times E_{res}(\text{secondary CH candidate}) + \frac{w_4}{d_{nextCH}} \tag{8}$$

whereby

$$w_1 + w_2 = 1 \tag{9}$$

$$w_3 + w_4 = 1 \tag{10}$$

The distance d_{nextCH} needs more clarification. The designation of the primary CH is carried out first. Then the CHs of neighbouring clusters are selected one at a time. This is done by moving to the right and left of the primary cluster (Vertical Clustering) or inwards and outwards (Spherical Clustering). Naturally, if the primary CH is found at an edge cluster, CH selection proceeds in one direction only. An edge cluster would be the leftmost or rightmost cluster in Vertical Clustering. In the case of Spherical Clustering, it would be the innermost or outermost cluster. Hence d_{nextCH} is the distance to the neighbouring CH which has already been elected.

The weights w1, w2, w3 and w4 are determined after extensive trial and error simulations. Weights w1 and w3 are varied from 0.1 to 1 and the combination of weights which result in the highest First Node Death (FND) is selected.

3.5 Proposed protocols

3.5.1 PEGASIS and PEG-GA. PEGASIS is extended to 3D by considering the third dimension for chain building by the greedy algorithm. Otherwise CH selection and transmission remain the same. PEG-GA is the same as PEGASIS except that genetic algorithm is used to build the chain.

3.5.2 Vertical clustering. Vertical Clustering used in two-dimensional networks by Ramluckun and Bassoo [10] is extended to three-dimensions which results in clusters in the form of cuboids rather than rectangles. This is achieved by clustering according to the value of x coordinate of nodes as shown in Figure 7. CH selection and data transmission occur similarly to the 2D equivalent. The CH mechanism proposed in Section 3.4 is used. A chain is built in each cluster using GA and data is routed to each CH similar to Pegasis. Then, multi-hop inter-cluster transmission takes place left to right and right to left until the primary CH is reached. The latter transmits to the BS.

Mathematically, the threshold for the C^{th} cluster is given by:

$$\frac{(C-1)M}{C_{total}} < x < \frac{CM}{C_{total}} \tag{11}$$

where C is the C^{th} cluster, C_{total} is the total number of clusters and x is the x coordinate of nodes.

3.5.3 Spherical clustering. Spherical Clustering is carried out according to the distance of nodes from the centre of the network. We consider an imaginary sphere of radius equal to the distance of the furthest node from the centre of the network. The sphere has centre (M/2, M/2, M/2) and is divided radially. Nodes are assigned a cluster according to between which radii of

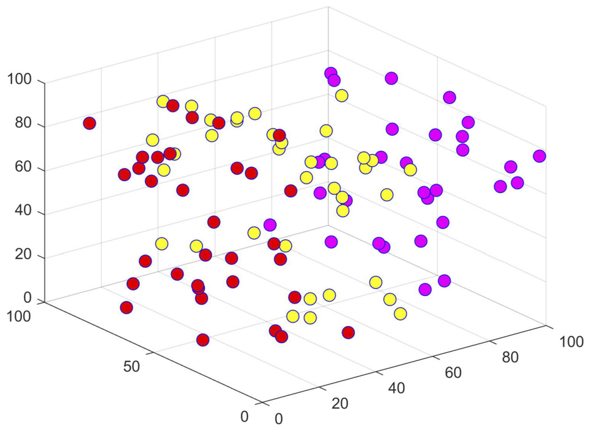

Figure 7.
Vertical Clustering (3 clusters).

the sphere they lie as illustrated in Figure 8. CH selection and data transmission occur similarly to Vertical Clustering except that multi-hop transmission occur inwards or outwards until primary CH is reached.

Mathematically, the threshold for the C^{th} cluster is given by:

$$\frac{(C-1)R}{C_{total}} < r < \frac{CR}{C_{total}} \tag{12}$$

where R is the distance from the centre of the furthest node and r is the distance of nodes from the centre.

3.5.4 Number of cluster selection. The main objective of clustering is to decrease the high delay created by a large number of hops present in a long chain. Increasing, the number of clusters in three-dimensional networks causes the number of inter-cluster communication to generally increase. Consequently, the energy consumed per round also increases. Therefore, a trade-off between delay and energy exhausted must be made. The average energy consumed per node per round for the first 1000 rounds and the maximum delay is considered.

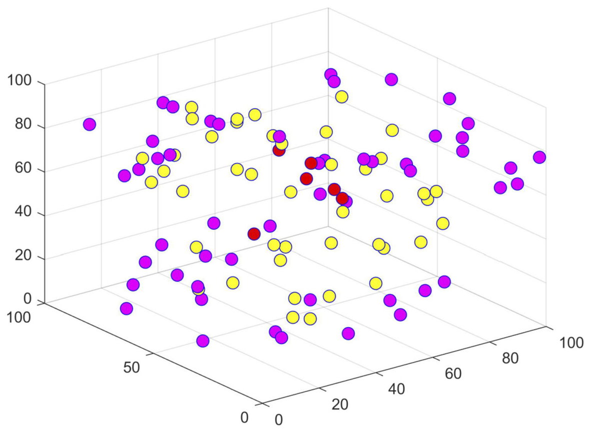

Figure 8.
Spherical Clustering (3 clusters).

The total energy consumed is:

$$E_{consumed} = \sum^{N} \text{initial energy} - \sum^{N} \text{residual energy} \tag{13}$$

$$= N \times E_o - \sum^{N} \text{residual energy} \tag{14}$$

The average energy consumed per node per round for the first 1000 rounds is given by:

$$E_{avg-1000} = \frac{E_{consumed} \text{ at } 1000 \text{ rounds}}{N \times 1000} \tag{15}$$

3.5.5 Flowchart of proposed clustered protocol. Figure 9 indicates the main steps taking place in the proposed clustered protocol.

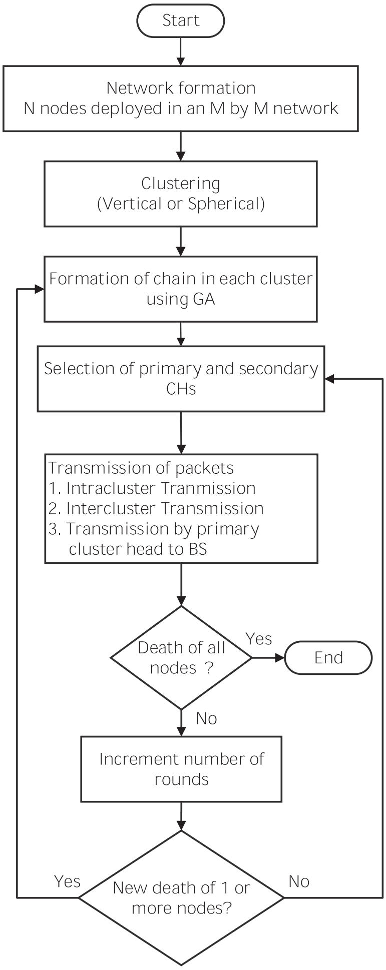
Start
Network formation
N nodes deployed in an M by M network
Clustering
(Vertical or Spherical)
Formation of chain in each cluster using GA
Selection of primary and secondary CHs
Transmission of packets
1. Intracluster Tranmission
2. Intercluster Transmission
3. Transmission by primary cluster head to BS
Death of all nodes ?
Yes
End
No
Increment number of rounds
New death of 1 or more nodes?
Yes
No

Figure 9.
Main stages in proposed clustered protocol.

3.5.6 Summary of proposed protocols. The main features of the proposed protocols have been summarised in Table 1.

3.6 Metrics

3.6.1 Lifetime. Network lifetime is an important metric which is concerned with the round number at which a certain number of nodes still have energy left. It has been given a number of definitions [23]. The following ones are considered in our study:

1. First time instant whereby any sensor node fully exhaust its energy. This is also known as stability period or First Node Death (FND).
2. Time instant when a percentage of nodes remain operational in the network. When the 50% mark is considered, it is referred to as Half Node Death (HND).
3. Time instant at which all nodes are dead commonly referred to as Last Node Death (LND).

As FND is the most important metric as many applications require all nodes to be alive to function properly, the analysis of the results will heavily focus on it.

To illustrate the performance of our protocols in relation to this metric, a graph of number of nodes alive against round number is plotted. Also, a table is provided to show the rounds at which FND, HND and LND occur (Table 2).

3.6.2 Residual energy. Residual energy indicates the total amount of energy that remain with the whole network. A graph of residual energy against round number is plotted in our results. It helps to visualise the overall energy consumption by the network at different points in time. The slope of the graph gives an indication of how much energy is spent at a particular point in the lifetime of the network. An increase in the slope would indicate that energy is expended more rapidly. As the lines on the graph may be very close to each other, two graphs

Routing Protocol	Chain Building	Clustering	CH Selection
PEGASIS	Greedy Algorithm	None	Round robin method
PEG-GA	Genetic Algorithm	None	Round robin method
PEG-GA Vertical Clustering	Genetic Algorithm	Vertical	Energy-Distance (Section 3.4)
PEG-GA Spherical Clustering	Genetic Algorithm	Spherical	Energy-Distance (Section 3.4)

Table 1. Main features of proposed protocols.

Parameter	Value
N (number of nodes)	100
M (size of network)	100
Coordinates of BS (Inside)	(50,50,50)
Coordinates of BS (Outside)	(50,50,300)
k (packet size)	2000 bits
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/m^2
ϵ_{emp}	0.0013 pJ/bit/m^4
E_{da}	5 nJ/bit/signal
E_0 (initial energy)	0.5 J
C	1 ms
Population size	100
Number of iterations	10000

Table 2. Network and GA Parameters.

are presented. One graph will show the residual energy over the whole lifetime and the other will be an enlarged version.

3.6.3 Load balancing. Load balancing is concerned with how well the routing protocol can distribute the energy consumption equally among the different nodes. It can be assessed using the percentage of residual energy of the whole network at FND [24]. A lower percentage of residual energy at FND indicates better load balancing.

3.6.4 Delay. The delay is given by (Fei, et al., 2016; Ammari & Das, 2005; Ammari & Das, 2008 cited in [10]):

$$Delay = (T_q + T_p + T_d) \times hops \tag{16}$$

$$= c \times hops \tag{17}$$

where is T_q the queue delay per intermediate forwarder, T_p is the propagation delay, T_d is the transmission and hops is the number of hops from first node transmitting (at the end of a chain) to the BS. The maximum delay of each routing protocol will be considered in the results section.

3.7 Assumptions and simulation parameters

The following assumptions were made during simulation:

1. Nodes transmit k-bit data packets at regular interval (called round).
2. Nodes are assumed to have global knowledge of the network.
3. The sink performs clustering and uses GA for chain building and informs the nodes about their neighbours.
4. All nodes are capable of sending data to the BS.
5. It is assumed that the energy required for transmission from one node to another is the same in both directions. This is known as a symmetric channel.
6. Control frames have a much smaller size than data frames. Hence, the energy expended in the transmission of control frames is negligible compared to that expended in the transmission of data frames and is therefore neglected.

4. Results and discussion

4.1 Cluster size choice

Figures 10 and 11 show the maximum delay incurred by the proposed clustered protocols for different number of clusters. These are valid for both BS inside and outside scenarios as the clustering is the same for both protocols with the weight for CH selection being the distinction.

As has been explained previously in Section 3.5.4, $E_{avg-1000}$ generally increases when increasing the cluster size as depicted by Figures 12–15. Two cluster size from each protocol was chosen, one which was more focused on minimising delay and the other one aimed at minimising the average energy consumed per node per round. This is shown in Table 3. The cluster size will be indicated in brackets for the clustered routing protocols. For instance, PEG-GA Spherical Clustering(3) indicates spherical clustering with three clusters.

4.2 BS inside

Figure 16 and Table 4 show that PEG-GA and PEG-GA Vertical Clustering(3) perform best with the latter outperforming the former by 31 rounds. PEGASIS shows very poor

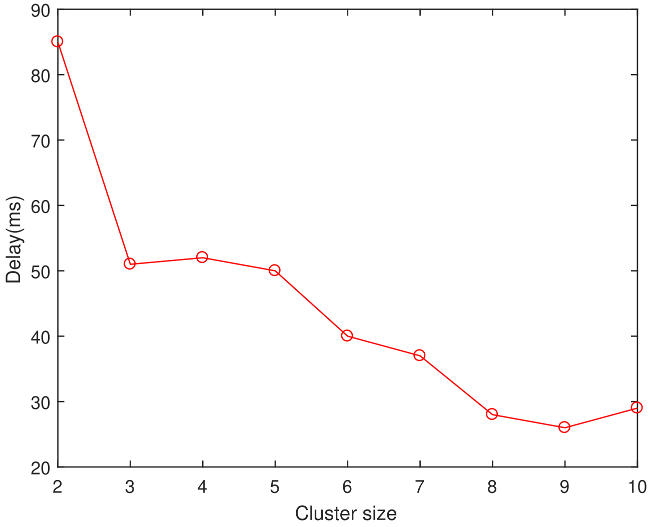

Figure 10. Maximum delay in PEG-GA Spherical Clustering with varying cluster size.

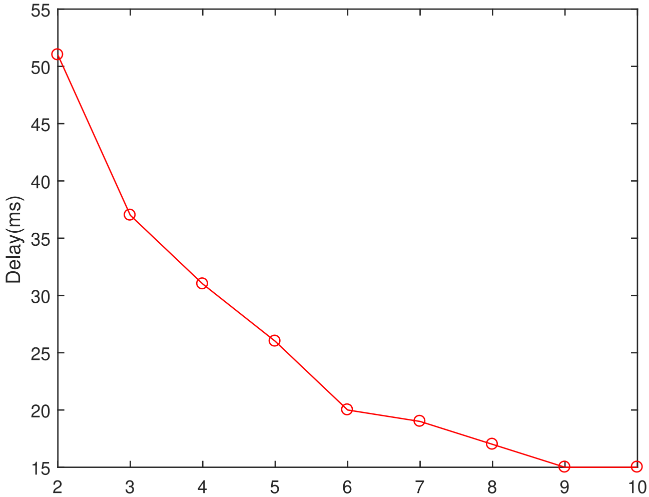

Figure 11. Maximum delay in PEG-GA Vertical Clustering with varying cluster size.

performance in terms of stability period, being inferior to PEG-GA Vertical Clustering(3) by 817%. This remarkably high percentage of improvement obtained is due to the greedy algorithm performing even worse in 3D networks relative to 2D ones. Vertical Clustering variants outperform their Spherical Clustering equivalent.

From Figures 17 and 18, it can be seen that PEG-GA performs best in terms of residual energy having the least steep slope. PEG-GA expends less energy as it does not incur losses in inter-cluster transmission as compared to the other clustered protocols. PEGASIS performs the worse as the greedy chain results in a non-optimum chain length. It is important to reiterate that the addition of a third dimension accentuates the difference

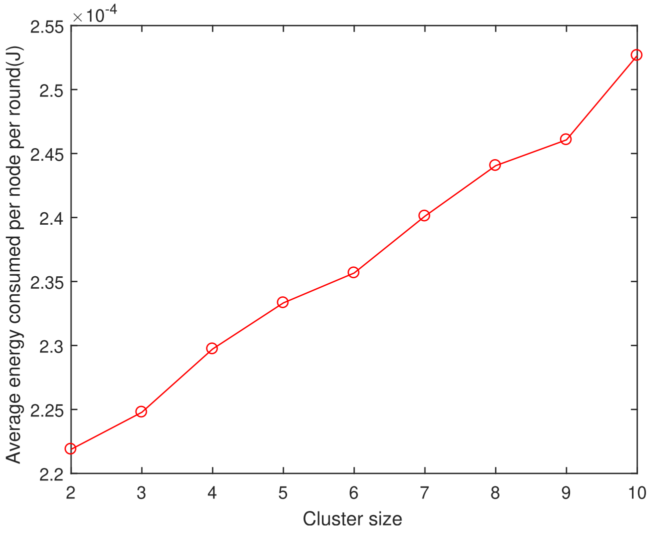

Figure 12.
Average energy spent in PEG-GA Spherical Clustering with varying cluster size(BS Inside).

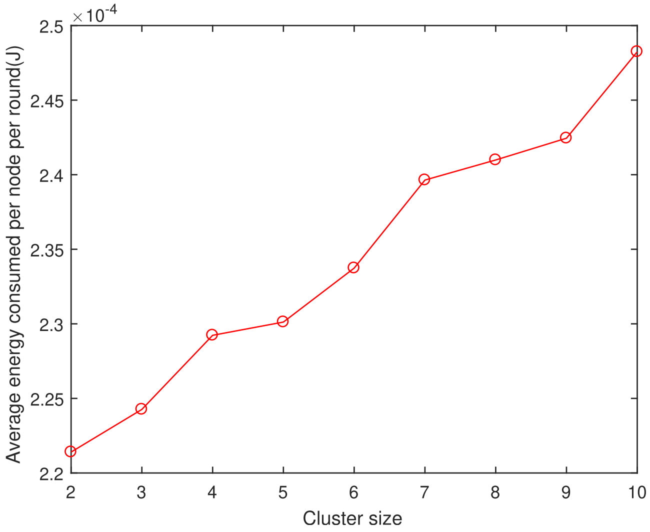

Figure 13.
Average energy spent in PEG-GA Vertical Clustering with varying cluster size(BS Inside).

between the greedy and the GA chain length explaining the 804% increase in FND from PEGASIS to PEG-GA.

It can be deduced from Table 5 that PEG-GA Vertical Clustering(3) performs best in terms of load balancing. PEGASIS and PEG-GA have the worst performance as CH are selected in a round robin fashion. PEG-GA Spherical Clustering(8) and PEG-GA Vertical Clustering(8) result in some clusters having a few number of nodes. The latter have to share the burden of inter-cluster transmission amongst themselves. Compared to clusters with considerably more nodes, the clusters with fewer nodes must use the same nodes for inter-cluster transmission more often, thus affecting load balancing in the whole network.

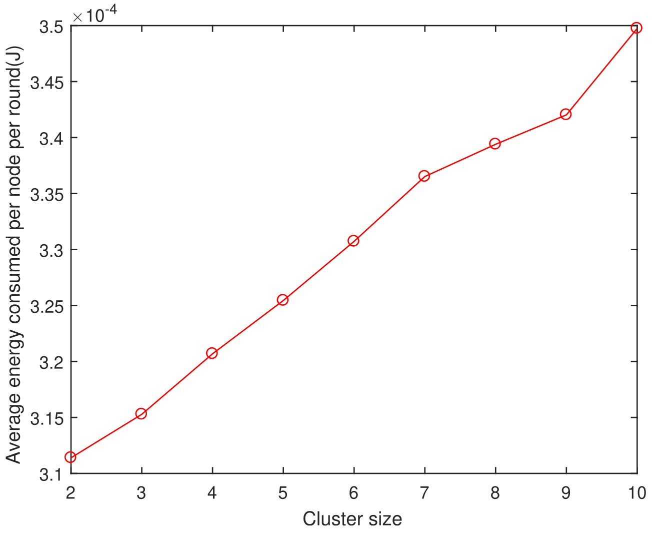

Figure 14. Average energy spent in PEG-GA Spherical Clustering with varying cluster size(BS Outside).

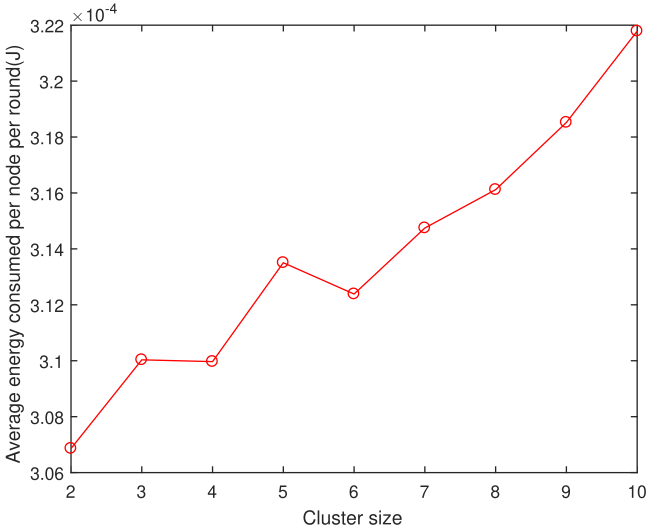

Figure 15. Average energy spent in PEG-GA Vertical Clustering with varying cluster size(BS Outside).

Routing Protocol	Cluster Size	
	BS In	BS Out
PEG-GA Spherical Clustering	3 and 8	3 and 8
PEG-GA Vertical Clustering	3 and 9	3 and 9

Table 3. Number of clusters selected.

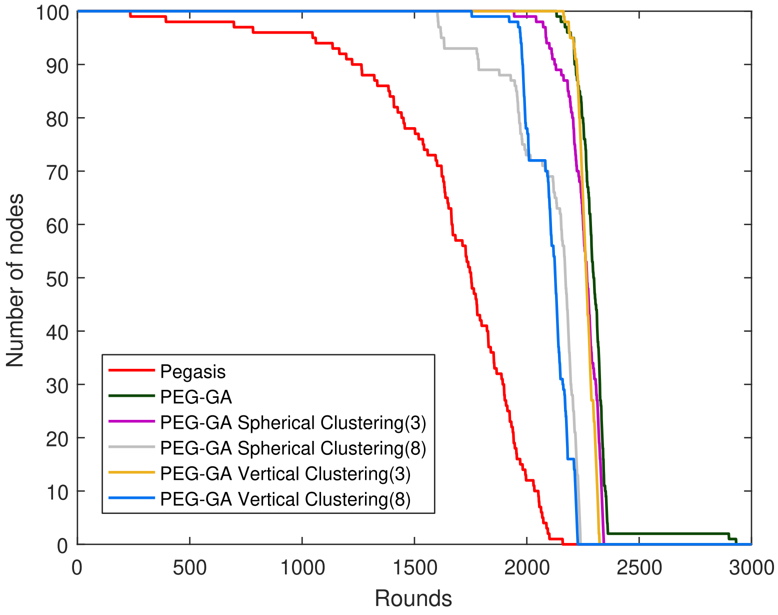

Figure 16.
Alive nodes vs rounds (3D BS Inside).

Table 4.
Lifetime (3D BS Inside).

Routing Protocol	FND	HND	LND
PEGASIS	236	1752	2160
PEG-GA	2133	2297	2932
PEG-GA Spherical Clustering(3)	1944	2268	2343
PEG-GA Spherical Clustering(8)	1603	2172	2239
PEG-GA Vertical Clustering(3)	2164	2265	2324
PEG-GA Vertical Clustering(8)	1755	2127	2227

4.3 BS outside

Figure 19 and Table 6 once again show the very poor performance of PEGASIS in a 3D environment as explained before and will result in the GA protocols exhibiting very high percentage of improvement over it in terms of FND. Contrary to the BS inside case, PEG-GA is not one of the better performing protocols in terms of FND. This is because of the method of CH selection which is completely energy and distance oblivious. The CH has a high transmission burden and will hence expend considerably more energy than in the BS Inside case. If a CH's energy level is low, this high energy expense is likely cause it to die. PEG-GA Vertical Clustering(3) outperforms other protocols for FND with a 420% and a 42.7% improvement over PEGASIS and PEG-GA respectively.

Figures 20 and 21 indicate that PEG-GA Vertical Clustering(3) is superior in terms of residual energy. As it has only 3 clusters, it incurs less energy expense for inter-cluster transmission. Additionally, it has a smaller overall chain length than the other protocols. In three-dimensional networks, the chain length of PEG-GA is smaller than the overall chain length in the clustered protocols. PEG-GA also does not have to cater for inter-cluster transmission. Nevertheless, it is outperformed by the three of the other clustered protocols due to the absence of a proper CH selection mechanism. PEG-GA Spherical Clustering(8),

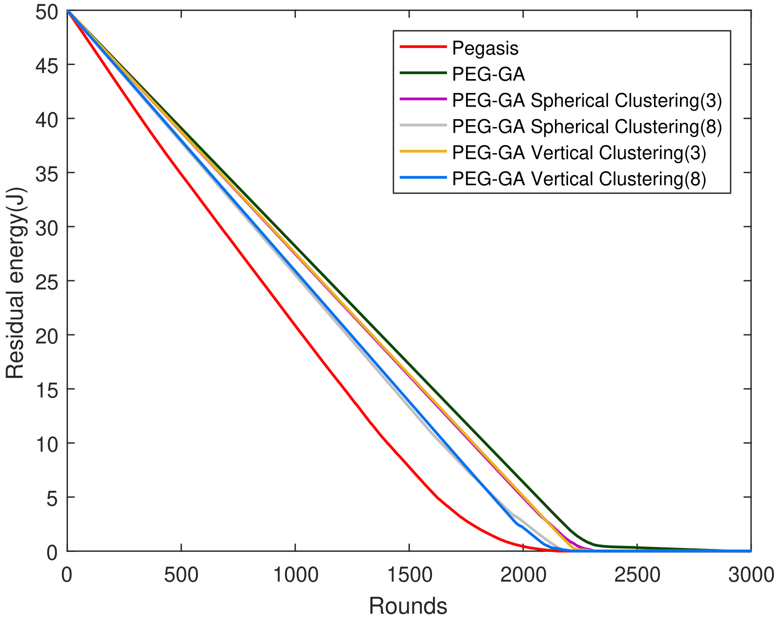

Figure 17.
Residual Energy vs rounds (3D BS Inside).

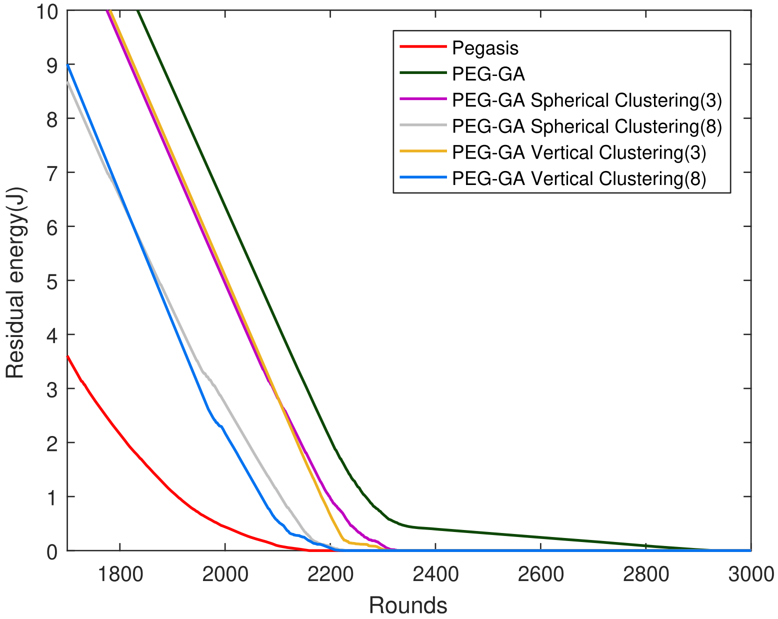

Figure 18.
Residual Energy vs rounds enlarged (3D BS Inside)[Enlarged].

despite having said selection mechanism, suffers from a long overall chain length. It can be observed that Vertical Clustering is more adept than Spherical Clustering at producing a smaller overall chain length for this particular network size and density.

Table 7 illustrates that PEG-GA Vertical Clustering(3) has the best load balancing property. The Spherical Clustering protocols are inferior to their equivalent Vertical Clustering protocols because of the uneven distribution of nodes in clusters. This causes the same nodes to be more frequently used for inter-cluster transmission in some clusters. This

Table 5.
Percentage residual energy at FND (3D BS Inside Case).

Routing Protocol	Residual Energy(%)
PEGASIS	85.5
PEG-GA	7.0
PEG-GA Spherical Clustering(3)	12.4
PEG-GA Spherical Clustering(8)	21.6
PEG-GA Vertical Clustering(3)	2.8
PEG-GA Vertical Clustering(8)	15.4

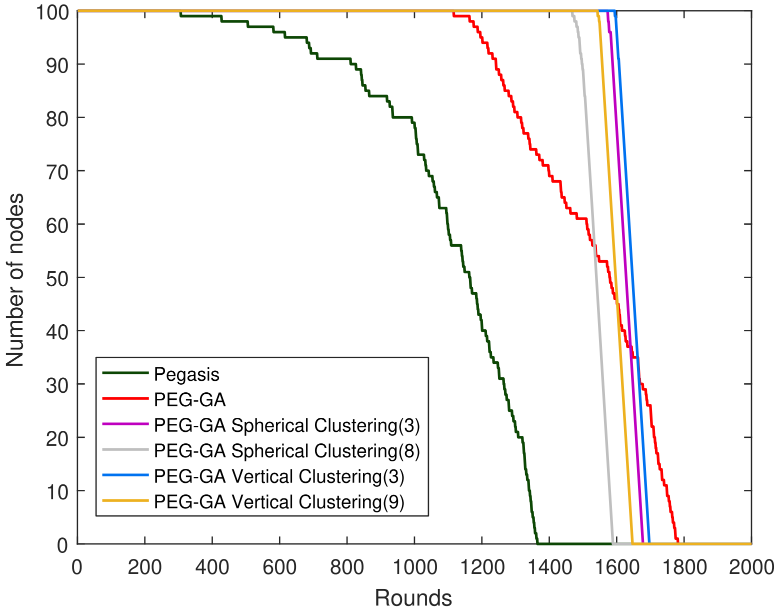

Figure 19.
Alive nodes vs rounds (3D BS Outside).

Table 6.
Lifetime (3D BS Outside).

Routing Protocol	FND	HND	LND
PEGASIS	307	1163	1365
PEG-GA	1117	1579	1782
PEG-GA Spherical Clustering(3)	1574	1628	1678
PEG-GA Spherical Clustering(8)	1469	1540	1589
PEG-GA Vertical Clustering(3)	1595	1647	1697
PEG-GA Vertical Clustering(9)	1544	1597	1647

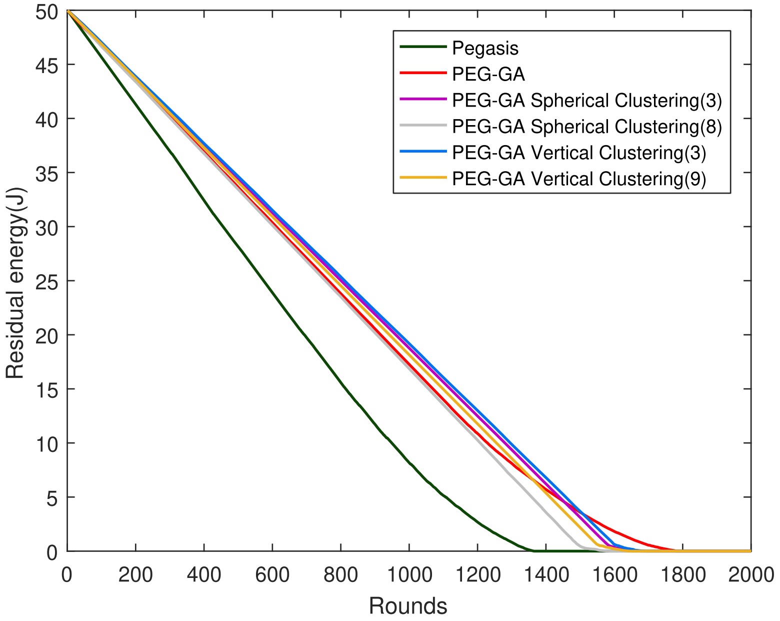

Figure 20.
Residual Energy vs rounds (3D BS Outside Case).

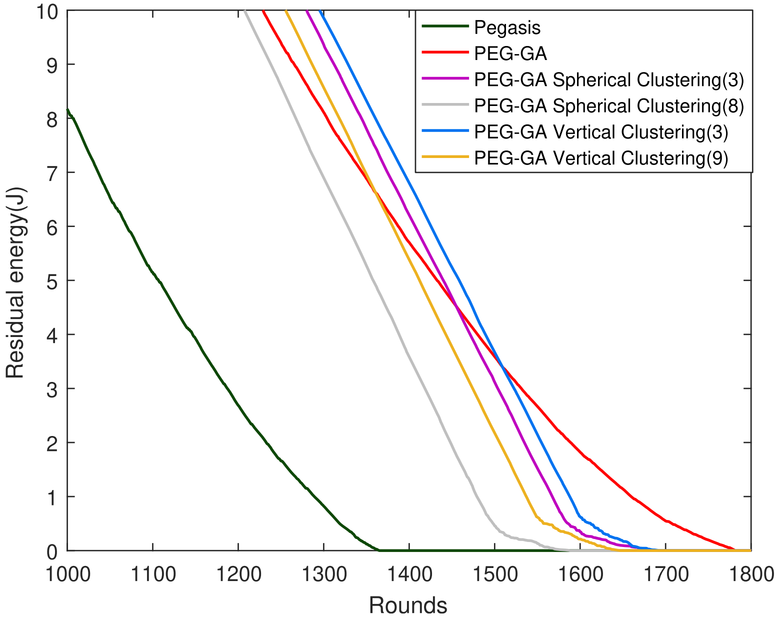

Figure 21.
Residual Energy vs rounds (3D BS Outside Case)[Enlarged].

Table 7.
Percentage residual energy at FND (3D BS Outside Case).

Routing Protocol	Residual Energy(%)
PEGASIS	73.5
PEG-GA	26.9
PEG-GA Spherical Clustering(3)	1.6
PEG-GA Spherical Clustering(8)	2.7
PEG-GA Vertical Clustering(3)	1.5
PEG-GA Vertical Clustering(9)	1.6

effect is more noticeable for a larger number of clusters with a larger difference of residual energy at FND (1.1%) observed between PEG-GA Spherical Clustering(8) and PEG-GA Vertical Clustering(9).

5. Conclusion and future works

In this paper, a novel method of combining PEGASIS, clustering and genetic algorithm to minimise transmission distance was proposed. A variable weight CH selection based on distance and energy for better load balancing was adopted. A new version of PEGASIS is adapted for 3D networks and variants of it were also proposed. Simulations in MATLAB have shown the superiority and adequacy of our proposed protocols in terms of FND. This is illustrated by PEG-GA Vertical Clustering(3) having improvements of 817% and 420% in FND over PEGASIS for base station inside and outside the network respectively. These very high percentages of improvement result from the very poor performance in 3D networks of the greedy algorithm used by PEGASIS and these percentages show the need for our proposed protocols. Residual energy, load balancing and delay performance were also shown to be superior.

Future research could make clustering to be more effective. Instead of building each chain separately after clustering, the whole problem could be formulated as the multiple travelling salesman problem and then a metaheuristic can be applied to solve it so as to minimise overall chain length further. Weights for CH selection are determined by trial and error. An algorithm like Particle Swarm Optimisation with a suitable fitness function could be developed to facilitate weight selection. Heterogeneous networks and mobility of nodes and base station are also areas that could be explored.

References

[1] M. Al Ameen, S.R. Islam, K. Kwak, Energy saving mechanisms for mac protocols in wireless sensor networks, Int. J. Distrib. Sens. Netw. 6 (2010) 163413.

[2] S. Jabbar, R. Iram, A.A. Minhas, I. Shafi, S. Khalid, M. Ahmad, Intelligent optimization of wireless sensor networks through bio-inspired computing: survey and future directions, Int. J. Distrib. Sens. Netw. 9 (2013) 421084.

[3] S. Lindsey, C.S. Raghavendra, PEGASIS: power-efficient gathering in sensor information systems, IEEE Aerospace Conference Proceedings 3 (2002) 1125–1130.

[4] X. Liu, J. Shi, I. Engineering, Clustering routing algorithms in wireless sensor networks: an overview, KSII Trans. Internet Inf. Syst. 6 (2012) 1735–1755.

[5] W. Guo, W. Zhang, G. Lu, PEGASIS protocol in wireless sensor network based on an improved ant colony algorithm, 2nd International Workshop on Education Technology and Computer Science ETCS 2010 (3) (2010) 64–67.

[6] T. Li, F. Ruan, Z. Fan, J. Wang, J.-U. Kim, An improved PEGASIS routing protocol based on neural network and ant colony algorithm, Int. J. Future Gener. Commun. Networking 8 (2015) 149–160.

[7] S. Ghosh, W. Bengal, Enhanced PEGASIS using Ant Colony Optimization for data gathering in WSN, in: 2016 International Conference on Information Communication and Embedded Systems (ICICES), 2016, pp. 1–6.

[8] S. Feng, B. Qi, L. Tang, An improved energy-efficient PEGASIS-based protocol in wireless sensor networks, in: Proceedings – 2011 8th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2011 4 (2011) 2230–2233.

[9] G. Zheng, Z. Hu, Chain routing based on coordinates-oriented clustering strategy in WSNS, in: 2009 International Symposium on Computer Network and Multimedia Technology, 2009, pp. 1–4.

[10] N. Ramluckun, V. Bassoo, Energy-efficient chain-cluster based intelligent routing technique for Wireless Sensor Networks, Appl. Comput. Inf. (2018).

[11] S. Roy, N. Mukherjee, Topology construction of 3d wireless sensor network, in: Advances in Computing and Information Technology, Springer, 2012, pp. 533–542.

[12] J. Heidemann, M. Stojanovic, M. Zorzi, Underwater sensor networks: applications, advances and challenges, Philos. Trans. R. Soc. A: Math., Phys. Eng. Sci. 370 (2012) 158–175.

[13] S. Kohli, P. Pratim Bhattacharya, M. Kumar Jha, Implementation of homogeneous leach protocol in three dimensional wireless sensor networks, Int. J. Sens. Wireless Commun. Control 6 (2016) 4–11.

[14] M. Baghouri, A. Hajraoui, S. Chakkor, Three-dimensional application-specific protocol architecture for wireless sensor, Networks 15 (2015) 352–360.

[15] H.H. Mukhairez, A.Y. Maghari, Performance Comparison of Simulated Annealing, GA and ACO Applied to TSP 6, 2015.

[16] D. Wohwe Sambo, B.O. Yenke, A. Förster, P. Dayang, Optimized clustering algorithms for large wireless sensor networks: A review, Sensors 19 (2019) 322.

[17] N. Anjum, M. Ahmad, S. Ahmad, Gateway based energy efficient routing: GEER, Int. J. Adv. Res., Ideas Innovations Technol. 3 (2017) 483–488.

[18] D. Sharma, A. Goap, A.K. Shukla, A.P. Bhondekar, in: Proceedings of 2nd International Conference on Communication, Computing and Networking, vol. 46, Springer Singapore, 2019, https://doi.org/10.1007/978-981-13-1217-5, URL: http://link.springer.com/10.1007/978-981-13-1217-5.

[19] Y. Jaradat, M. Masoud, I. Jannoud, A mathematical framework of optimal number of clusters in 3D noise-prone WSN environment, IEEE Sens. J. PP (2018) 1.

[20] M. Mitchell, An Introduction to Genetic Algorithms, MIT, 1998.

[21] D. Gupta, S. Ghafir, An overview of methods maintaining diversity in genetic algorithms, Int. J. Emerging Technol. Adv. Eng. 2 (2012) 56–60.

[22] J. Kirk, Traveling salesman problem – genetic algorithm, 2014. MATLAB Central File Exchange.

[23] H. Yetgin, K.T.K. Cheung, M. El-Hajjar, L. Hanzo, A survey of network lifetime maximization techniques in wireless sensor networks, IEEE Commun. Surv. Tutorials 19 (2017) 828–854.

[24] H. Kareem, H. Jameel, Maintain load balancing in wireless sensor networks using virtual grid based routing, in: Protocol, 2018 International Conference on Advanced Science and Engineering (ICOASE), 2018, pp. 227–232.

Corresponding author
Vandana Bassoo can be contact at: v.bassoo@uom.ac.mu