International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

Intelligent Deployment Strategy for Heterogenous Nodes to Increase the Network Lifetime of Wireless Sensor Networks

Riya Jindal
Computer Science and Engineering
Delhi Technological University
Delhi, India

Dr. Vinod Kumar
Assistant Professor
Department of Computer Science and Engineering
Delhi Technological University, Delhi

***

Abstract— We propose intelligent deployment strategies for heterogeneous node deployment in wireless sensor networks. Placing the nodes in a wireless sensor network is a difficult problem, so we use Particle Swarm Optimisation (PSO) algorithm and Genetic Algorithm (GA) to find the optimal location of the heterogeneous nodes thereby increasing the network lifetime. The routing protocols used in this implementation are direct routing protocol and multi-hop routing protocol. After performing many experiments, we found that the proposed methods increase the network lifetime on average up to 150% of the initial lifetime. The objective function used by the models is the inverse of a lifetime and we tracked lifetime and time taken to reach a solution for comparing GA and PSO.

Keywords— GA, PSO, heterogeneous nodes, wireless sensor network, network lifetime

I. INTRODUCTION

Wireless Sensor Networks have developed into a large study field in the last decade, introducing a wide range of fascinating new applications and recently becoming a vital part of the Internet of Things idea. A WSN is made up of a large number of small low-power sensor nodes that can detect a variety of physical phenomena (such as motion, light, sound, seismic activity, temperature, and so on) in nearly any sort of environment (military, volcanic, industrial, etc.). Through multi-hop routing, these sensor devices, known to simply as sensors, digest the raw sensory data and send it wirelessly to one or more data assembly tanks, known as sinks. For the purposes of processing and querying the acquired data, the sink(s) are linked to another wireless or wired network.

Nonetheless, WSN's enormous promise in critical applications is equaled by their extremely difficult design process. Because of the significant energy, computation, and networking limits of accessible sensor devices, WSNs are essentially distinct from traditional wired and wireless networks. WSNs are also very application-oriented, necessitating flexible configurations and cross-layer communication protocol stack optimisations. As a result, nearly every area of WSN architecture, such as MAC, data routing, and transport protocols, has been extensively researched.

Another important design consideration is sensor deployment, which is often referred to as sensor positioning, deployment, layout and placement in the literature. We shall use the phrase deployment in this survey. The deployment of a WSN has an impact on practically all of its performance measures, including network's lifetime, sensor connection and network's successful coverage. In general, there are two types of WSN deployment techniques: random deployment and planned deployment. Sensors are often spread (e.g., via airplanes) in random deployment, resulting in a stochastic dispersion of sensors, however their concentration can be regulated to some extent. In some implementations where the field of interest is unreachable, for example disaster zones and live conflict zones, random deployment may be the only viable option.

Planned deployment, on the contrary, is described as selecting sensor placements to optimise one or more WSN design objectives within the limits of a given application. Optimising coverage, limiting power consumption (i.e. maximising network longevity), and ensuring good network connections are all frequent design objectives. Planned deployment is compatible with a huge range of WSN applications, as long as the AoI is available. Border observation, facility access control, and architectural maintenance are all examples of this type of monitoring. If the implanted sensors are mobile, it can even be done in unreachable AoIs after a first random deployment.

In this paper, we offer a new categorisation of the approaches and algorithms for planned deployment of WSNs which have been suggested in the research. The mathematical methodology utilised to model and solve the deployment issue is the basis for our categorisation. Genetic Algorithms (GA) and Particle Swarm Optimisation (PSO) are two separate mathematical methodologies described.

It's worth noting that the computational techniques in our categorisation are heuristics, capable of offering sub-optimal solutions for most WSN applications. For many planned deployment challenges, traditional deterministic optimisation approaches capable of generating optimal solutions, such as Linear Programming, are ineffective. This is because a pragmatic planned deployment situation might have many design goals, heterogeneous WSNs, or/and a high percentage of sensor nodes. Such issues are known as NP-hard (non-deterministic polynomial time) problems, since no polynomial-time methods are able to solve them fully. This shows that the time it takes to obtain optimal solutions for these issues using deterministic optimisation methods, without applying any approximation methods, grows rapidly as their size expands.

II. SENSING MODEL

In general, a sensing model for a certain type of sensor is a statistical model that explains the sensor's likelihood of detecting an event or target. The probability of sensing an event by a sensor (s_i) is given by P_{ij}, which is a function of various factors, assuming the event occurs at a point p_j in the Area of Interest. The Euclidean distance (d_{ij}) between them, the sensor's alignment (for example, vision sensors with a specific Range of Vision, and different ambient factors are the most often utilised parameters). There exists a variety of sensing models in research. However, probabilistic and binary sensing models can be widely categorised.

A. Binary Sensing Model

This model implies that a sensor's detecting range (r_s) is fixed. When an event happens at a point that is less than or equal to the detecting range from position of the sensor, the event is probabilistically recognised by the sensor. If the distance is greater than the detecting range, however, the event will not be recognised. A binary sensing model is mathematically given as :

$$C_{xy}(s_i) = \begin{cases} 1 \text{ if } d(s_i, \rho) < r_s \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

where $C_{xy}(s_i)$ shows that point ρ is covered by s_i and $d(s_i, \rho)$ represents the Euclidean distance amid point ρ and sensor node s_i.

Because of its simplicity, this model is extensively used in research, although it is impractical. It's improbable that the effective sensing, or the signals of the observed event or target, will decline from max to nil in an instant. As a result, adopting a binary sensing model could result in sensors being under-utilised and, as a result, more sensors being deployed than needed, resulting in greater deployment costs.

B. Probabilistic Sensing Model

The goal of probabilistic sensing models is to incorporate the numerous factors that influence sensor reading accuracy. These variables include ambient circumstances such as noise and obstructions, in addition to the nature of the observed physical phenomena and the sensor's poor detection potential.

Let r_e represent the detection uncertainty, with $0 < r_e < r_s$ and Eq. 2 as the mathematical representation. The expressions $r_s - r_e$ and $r_s + r_e$ define a ring in which an event may or may not be sensed depending on the coverage probability value.

(2)

where, r_e represents the uncertainty and bounded by

$0 < r_e$ and $< r_s$

$$C_{xy}(S_i) = \begin{cases} 0, & \text{if } r_s + r_e \le d(s_i, p) \\ e^{-\delta a^{\beta}}, & \text{if } r_s - r_e < d(s_i, p) < r_s + r_e \\ 1, & \text{if } r_s - r_e \le d(s_i, p) \end{cases}$$

and

$$a = d(s_i, p) - r_s - r_e$$

where, β *and* δ measure the probability of detection.

III. Sensor Mobility

WSNs are divided into two categories based on the nature of sensor nodes : static WSNs (WSN) and mobile WSNs (MWSNs). Recent advancements in cloud technology and robotic technologies have sparked interest in MWSNs. WSNs that include sensing, communication, computing, and transportation capabilities are referred to as MWSNs. Installing static sensors on mobile carriers can provide transportation capabilities. The MWSN could be homogeneous, that is, it only comprises mobile sensors, or heterogeneous, that is, it includes both static & mobile sensors.

The increased mobility can bring numerous benefits to the MWSNs, but it can also considerably complicate the layout. The capability of the network to self-deploy after a randomly initialised deployment is the major additional benefit. Due to variables like wind, vegetation, topographical imperfections, and other considerations, this self-deployment (or, more precisely, re-deployment) capacity can greatly enhance the network's effective coverage from the original restricted coverage that is difficult to regulate. In heterogeneous WSNs with primarily static sensors, mobile sensors may also be utilised to fill coverage gaps in an AoI. Another significant advantage is the network's capacity to self-reconfigure. If some of the network nodes die (due to power exhaustion or environmental stress), resulting in connectivity islands, reconfiguration could be highly advantageous. Reconfiguration will allow the network to maintain adequate connection, allowing multi-path routing to continue.

Unfortunately, the added mobility introduces design issues such as a heavy load on the generally battery-powered sensors' limited power resources and the requirement for coordination amongst the mobile sensors.

IV. Mathematical Techniques Used In WSN Deployment Algorithms

We present two mathematical methods widely utilised for constructing re-deployment and planned deployment algorithms for WSNs. We hope to acquaint the reader with the origins and fundamentals of these mathematical techniques in this part.

A. Genetic Algorithms (GA) :

GAs are optimisation and search algorithms based on biological evolution. Since John Holland's work in the early 1970s, GAs have been used to solve optimisation challenges in a variety of disciplines, including communication networks, manufacturing engineering, and artificial intelligence. The GA paradigm is based on Darwin's theory of natural selection, which states that "species whose individuals are best adapted survive; others go extinct." In stochastic & multi-objective optimisation problems, when deterministic optimisation approaches are ineffective, a GA can be very useful. A GA has three fundamental components :

- A genetic description of the problem's potential solutions. This is known as encoding, and it is based on a problem's constraints and variables. Candidate solutions are encoded in a way that they can be decoded into a distinct variables' array that belongs to the search space, allowing the constraints to be verified. Integer encoding, binary encoding, and real number encoding are some of the encoding methods used in GAs. The type of encoding method to apply is largely determined by the nature of the optimisation problem. The phenotypic space contains possible solutions in the problem's search space, whereas the genotype space contains their genetic representation through encoding.
- For evaluating potential solutions, a fitness function is used.
- During the reproductive phase of the GA, probabilistic genetic operators change the constitution of the offspring.

In one iteration of a standard GA, five steps are performed. The first step is to create a population of chromosomes or individuals to work with. Each chromosome represents a distinct possible solution to the problem, encoded in a different way. The problem's search space is generally uniformly covered by the initial population.

Following the creation of the initial population, step two is completed, which involves evaluating the individuals in the population using a fitness function. The fitness function is a mathematical representation of what we wish to maximise, and is basically a cost function. Fitness evaluation is used by GAs to weed out the weakest members of the population and identify the fittest members. As a result, a chromosome is considered to be the fittest if it gets the fitness evaluation closest to the optimal point than the others.

Parent selection is the third stage, which involves choosing chromosomes from the population to proceed through the GA's reproductive stage. Parent selection is frequently stochastic and is generally based on computed fitness. The Roulette Wheel and the Tournament methods are the two most often utilised parent selecting procedures.

To create an offspring or children population, the fourth step is to apply the two genetic operators, mutation and crossover, to the chosen parent chromosomes. Crossover is the most common genetic operator, and it works by pairing every two parents (individuals) in the population at random to create children with parts of both their codes. Mutation, on the contrary, is a covert operation that generates a new entity by changing a randomly picked element of a chosen parent. Both operators are in charge of steering an offspring population into new areas of the search space for the problem. The same fitness criteria are used to assess the offspring population.

The selection phase is the fifth stage in a GA, and it involves selecting individuals from both the parent and offspring populations to create a new population. The GA's driving force is selection, which directs the search to favourable areas of the search space. There are numerous stochastic and deterministic selection approaches, such as fitness-based selection, age-based selection and elitism. Steps two through five are performed numerous times to generate generations, and the algorithm eventually converges to the fittest individual, who should represent the best possible solution, but this isn't guaranteed. The process can either end after a certain number of generations have been produced or after discovering an individual with a fitness that corresponds to a reasonable solution to the problem. Table I shows the overall structure of GAs in pseudo code. The capacity of GAs to cope with multi-objective optimisation and combinatorial issues is one of its most significant advantages as an optimisation technique. GAs were used in the development of multi-objective deployment algorithms for WSN because of this useful trait.

TABLE I
PSEUDO CODE OF A GENERAL GENETIC ALGORITHM

Step	Genetic Algorithm
1.	Set $t \leftarrow 0$
2.	**Initialize** $P(t)$
3.	**Evaluate** $P(t)$
4.	**While** (termination condition not met)
5.	**Recombine** $P(t)$ to yield $C(t)$
6.	**Evaluate** $C(t)$
7.	**Select** $P(t+1)$ from $P(t)$ and $C(t)$
8.	$t \leftarrow t+1$
9.	**End While**

B. Particle Swarm Optimisation (PSO) :

Swarm intelligence is a field of AI that studies the cohesive behaviour and characteristics of complicated, self-organised, distributed systems with a social structure, such as fish schools, ant colonies and bird flocks. This network is made up of basic interacting agents that are grouped into tiny societies called swarms and display intelligence characteristics including the capacity to react to environmental challenges and make decisions.

Swarm intelligence was included in the optimisation framework in the form of a collection of algorithms for controlling robotic swarms that were first described in

TABLE II
PSEUDO CODE OF PARTICLE SWARM OPTIMIZATION

Step	Particle Swarm Optimization
1.	Set $t \leftarrow 0$
2.	**Initialize** S and **Set** $M \equiv S$
3.	**Evaluate** S and M; **Define** index g for best position
4.	**While** (termination condition not met)
5.	**Update** S using (18) and (19)
6.	**Evaluate** S
7.	**Update** M; **Redefine** index g
8.	$t \leftarrow t+1$
9.	**End While**
10.	**Print** best position found

1989. Three key swarm intelligence optimisation methods, Stochastic Diffusion, Ant Colony Optimisation, and Particle Swarm Optimisation, were developed a few years later. Here, we'll solely discuss PSO in this work since it is becoming more widely used in the development of WSN deployment algorithms.

PSO, a randomised global optimisation mechanism based on social modelling techniques, was created by Eberhart and Kennedy in 1995. The PSO algorithm's fundamental concept is to utilise a population (swarm) of search points (particles) that move probabilistically in the search space of an optimisation problem. The best solution (i.e. the best position) ever attained by each individual in the population is stored in memory as experience. This information is then sent on to a portion or all of the colony, steering its motion towards the parts of the search space where it is most likely to discover the best answer. The selected communication scheme has a significant impact on the algorithms' convergence.

V. Wireless Sensor Networks Deployment Algorithms

We now review these algorithms, including their performance, assumptions, goals, using classification of the complex analytical approaches used in WSNs deployment algorithms outlined in section IV.

A. *Genetic Algorithm*

Genetic Algorithm is based on Darwin's theory of evolution or more importantly on the theory of survival of the fittest. The basic building block of any genetic algorithm is called a chromosome. And a chromosome is a set of variables that could potentially represent a solution to the problem. In our case, a chromosome contains a list of locations to which heterogeneous nodes are to be deployed. The first step in this algorithm is to create a population. Now each chromosome in a population could be a potential solution to our problem. The next step is to perform parent selection. Parents are selected based on their fitness values and the parents with the best fitness, move on to the next generation. Next step is to perform crossover. Crossover is a process in which we share genetic information among parents so that we could generate best candidates or a best solution in the next generation. The last step in performing Genetic Algorithm is called mutation. Mutation is a process in which we mutate the current chromosome so that it could give better solution in next generation. Lastly these steps : selection, crossover and mutation are performed iteratively until a solution is reached.

B. *Particle Swarm Optimisation*

Dr. Kennedy and Dr. Eberhart developed Particle Swarm Optimisation Algorithm in 1995, based on the social behaviour of fish schooling and birds flocking. PSO shares common routes with other similar metaheuristic techniques like Ant-Colony optimisation, Genetic Algorithm, and Cuckoo's Search. It is very similar to evolutionary computation technique such as Genetic Algorithm such as the system starts with a population of random solutions and then updates generations to look for optima. In contrast to GA, however, PSO lacks evolution operators such as crossover and mutation. Particles, which are possible solutions in PSO, follow the current optimal particles through the problem space. When compared to GA, PSO has the advantages of being simple to implement and having few parameters to modify. The velocity and convergence rate are all decided by the factors like inertia term, cognitive component and social component. The fitness value calculated for each individual particle is stored in the vector called eBest and the global best which is known as global optima for every particle is stored in globalBest. My main objective is to increase the lifetime of wireless network sensors by adding heterogeneity.

So, let's see how PSO determines the best possible position in 2-Dimensional space. The most optimum solution is found by the PSO algorithm in which considering our nodes as "swarms" in the 2-Dimensional space, we find the optimal position and if any node is placed at a corresponding colony, then it's considered the best position for adding the heterogeneous node by taking in the factors like energy conservation. If a node doesn't exist at the coordinate prescribed, then the nearest node to that node is chosen. It takes 'n' iterations to find the global optimum. For implementing 'x' heterogeneous nodes, it takes x*n iterations to conclude the placement of all the heterogeneous nodes.

VI. Experimental Results

We tested GA and PSO on 500 X 500 terrain and 1000 X 1000 terrain.

1. 50% improvement in the result was observed in 500 X 500 terrain and up to 180% increase was observed in 1000 X 1000 terrain.

2. On a 500 X 500 terrain, both the PSO and GA have comparable performance, however on a 1000 X 1000 terrain PSO has performed better.

3. GA has always been faster than the PSO algorithm. PSO works better on a larger terrain.

4. However, there are limitations to this approach like increase in terrain size, increases time complexity and reducing the randomness of the algorithm can lead to better convergence. This approach can be further improved by employing different routing algorithms to conserve energy. Thereby, increasing the lifetime of the network. Since we are using evolutionary algorithms, we may not always find the best solution but it is guaranteed that there will be a good solution to the problem.

Conclusion

Because of the rapid development of smart cities, WSNs are playing an increasingly important part in our daily lives. When sensor nodes are dispersed and need to stay active to acquire and send data from one site to another, deployment challenges in WSNs become critical. In most cases, a large number of sensor nodes are placed in the operational area. As a result, attaining the lowest number of sensor nodes that can surround the whole field of interest is critical for research. This study examines the present state of the art in the domain of node deployment in WSNs. Genetic Algorithms and Particle Swarm Optimisation are the two techniques proposed for this categorisation. In addition, a detailed comparison of the advantages and pitfalls of these techniques is presented.

REFERENCES

1. L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in Proc. International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2011, pp. 1–6.
2. V. Potdar, A. Sharif, and E. Chang, "Wireless sensor networks: A survey," in Proc. International Conference on Advanced Information Networking and Applications Workshops (WAINA'09), 2009, pp. 636– 641.
3. J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," vol. 52, no. 12. Elsevier, 2008, pp. 2292–2330.
4. Deif, D. S., & Gadallah, Y. (2014). Classification of Wireless Sensor Networks Deployment Techniques. IEEE Communications Surveys & Tutorials.
5. I. Stoianov, L. Nachman, S. Madden, T. Tokmouline, and M. Csail, "Pipenet: A wireless sensor network for pipeline monitoring," in Proc. 6th International Symposium on Information Processing in Sensor Networks, (IPSN'06), 2007, pp. 264–273.