

A Genetic Algorithm Integrated Approach for Efficient Relay Nodes Placement

Sardar Anisul Haque¹, Mohammed Alreshoodi², Bader Alshaqqawi³, Ibrahim Alsukayti⁴ and Khalid Alsatami³

¹*Department of Mathematics and Computer Science, Alcorn State University, Lorman, Mississippi, USA.*

²*Department of Applied Science, Unizah Community College, Qassim University, Buraydah, Saudi Arabia.*

³*Department of Mathematics, Collage of Science, Qassim University, Buraydah, Saudi Arabia .*

⁴*Department of Computer Science, Collage of Computer, Qassim University, Buraydah, Saudi Arabia.*

ORCID: ¹0000-0002-3497-3156, ²0000-0002-3066-6909

Abstract

Relay Node Placement (RNP) is a critical design problem in Wireless Sensor Networks (WSNs). Network performance and energy efficiency can be highly affected by the placement strategy of RNs. It is an NP-hard optimization problem which can be effectively addressed with multi-objective formulation and solved using metaheuristics. The main aim of this research work is to efficiently optimize the unconstrained deployment of energy-harvesting RNs in a pre-established stationary WSN. The focus is on optimizing different conflicting objectives, namely network coverage, energy consumption, and deployment cost. This was approached in this paper using a novel integrated approach combining the Genetic Algorithm (GA) with a greedy technique. The greedy algorithm plays a vital role in the optimization process as it effectively guides GA during convergence. The outcome of the greedy algorithm is used to limit the search space complexity of GA and realize the critical places at which RNs might be placed. The greedy-integrated GA approach was evaluated extensively with different WSN scenarios, in comparison to the classic GA when used independently. The experimental results clearly demonstrate that the solutions to RNP problems, with considerably large search space and good number of sensors, can be significantly improved using the proposed placement strategy considering both solution quality and convergence time.

Keywords: Wireless Sensor Networks, Relay Node Placement, Genetic Algorithm, Greedy Algorithm

I. INTRODUCTION

Wireless Sensor Networks (WSNs) provide effective support for many applications in different domains. Common WSN applications include environmental sensing, surveillance, fire detection, health monitoring, battlefield reconnaissance, and natural disaster prevention. In such applications, a set of sensor nodes having computing, sensing, and communication resources is deployed in different terrain and environmental conditions. Sensor nodes are typically small-sized devices of limited processing and communication capabilities. They are commonly operated using batteries as power sources which would limit their lifespan. Moreover, the battery replacement

process is demanding in cost, time, and manpower. This makes energy efficiency of great significance for WSNs.

In wireless sensor networks, energy consumption is mostly counted for wireless radio communications. Therefore, low-power wireless communication technology has become widely utilized to minimize energy consumption and maximize network lifetime. Although it helps in reducing transmission power of sensor nodes, this approach comes at the cost of limiting network coverage in practice. Without addressing such limitations of WSNs, real-world deployments of the different WSN applications can be easily hindered. One strategic approach in this context is the deployment of Relay Nodes (RNs) in optimal locations to improve the performance of WSNs. RNs typically have larger energy capacity compared to sensor nodes as they would be mains-powered or energy-harvesting devices. Accordingly, RNs are costly components thus demand well-planned deployment for better cost management in regards to maintenance and deployment.

This challenging problem is known as the Relay Node Placement Problem (RNPP). It is classified as a Non-Deterministic Polynomial-time (NP)-hard optimization problem. Adopting approximate techniques to solve such a problem rather than exact techniques is more effective to maintain lower computing time as the problem dimension increases. Therefore, the RNPP can be efficiently addressed using metaheuristics. There are many metaheuristics among which evolutionary algorithms (EAs) are widely adopted due to their simplicity and effectiveness. multi-objective problems can be addressed using EAs which are capable of solving problems of different types without prior knowledge of the problem domain. The RNPP can be solved with Multi-Objective (MO) formulation using metaheuristics to optimize different conflicting objectives in simultaneous manner.

There are different considerations regarding routing structure, connectivity requirements, deployment strategies in the context of RN placement in WSN. Routing structure is classified into single-tiered and two-tiered structures. In the former, the network is structured in a star topology to connect each sensor node to at least one RN. The two-tiered structure organizes the network into multi-hop clusters such that RNs become the clusters' heads. For the connectivity requirements, RN placement can be accomplished according to the connected and survivable network models which ensure the connectivity and

bi-connectivity of the sensor nodes, respectively. In regards to deployment strategies, it is possible to carry out unconstrained deployment which enables anywhere RN placement. However, constraints which would be posed in some applications (e.g. limiting placement locations or physical distance) can also be considered when placing RNs.

In this paper, the problem was formulated as a two-tiered RNPP with connected and unconstrained deployment. We address efficient deployment of energy-harvesting RNs in pre-established WSNs with stationary sensor nodes. It was approached as a multi-objective NP-hard optimization problem using the Genetic Algorithm (GA) integrated with a greedy technique to optimize network coverage, energy consumption, and deployment cost. Integrating the greedy algorithm helps in limiting the search space for placing RNs which would lead the GA to better convergence time. From the experimental results, it is evident that the solutions to RNPPs, with considerably large search space and good number of sensors, can be improved significantly with this integration considering both the quality of solution and convergence time.

In the following section, a number of relevant research works are discussed. Section III presents the main assumptions considered for this work. In Section IV, the mathematical model is presented whereas the integrated approach is described in Section V. The evaluation setup is illustrated in Section VI in addition to a discussion of the evaluation results. Section VII concludes this research paper and gives a perspective on the future enhancements to the current work.

II. RELATED WORK

Metaheuristics, in particular GA, have been widely adopted to address different optimization problems in WSN. These include clustering [1], Routing [2], and node scheduling [3]. In recent studies, researchers carried out different surveys on different multi-objective problems found in WSNs [4-6]. It is also evident that RNPP was given considerable attention by the research community as different approaches using varying evolutionary algorithms were introduced [7-9]. Among these, GA was a popular method to approach the RNPP in an effective manner.

In [10], the RNPP was approached using a GA-based algorithm in addition to a greedy algorithm. The focus was on the deployment of the minimum number of RNs in a predetermined set of locations with the objective of establishing k -connectivity between each sensor node and RNs. It was found out that the GA-based approach outperformed the greedy placement strategy in different experimental setups. In a further research work [11], another GA-based solution was presented to factor in additional objectives including the network coverage of sensor nodes. As in both proposals, however, the RNPP was addressed without considering energy efficiency which is a critical objective to optimize network lifetime. The authors in [12] adopted the GA to address the RNPP considering the minimization of the number of RNs while maintaining maximum connectivity among sensor nodes and RNs. Other solutions have also been proposed to place RNs at a set of candidate positions. This approach is referred to as

Constrained RNPP (C-RNPP) [13]. It has been shown that this approach enables the placement of a minimum number of RNs in predetermined potential positions [14].

In order to facilitate effective multi-objective optimization for WSN, further modification and integration were applied to the GA in different research efforts. In [15], the GA was modified to incorporate a novel-sorting process. In [16], the proposed approach for solving the RNPP was based on the integration and cooperation of an exact method and metaheuristic. The GA was integrated with the Mixed-Integer Linear Programming (MILP) to support the genetic operations. The target was to optimize two main objectives namely end-to-end delay and network throughput. In [17], the GA was made more flexible by incorporating variable-length encoding and modifying the crossover and mutation operations. This approach enables fine-grained deployment while flexibly adjusting the number and position of the nodes. In [18], a combination of evolutionary algorithms and the concept of quantum computation was considered to minimize computational complexity. A GA-based solution inspired by quantum principles was proposed to address the RNPP considering two-tiered WSN architecture. It provides a solution for ensuring full coverage of sensor nodes using the minimum number of RNs while having connectivity established among the RNs at the top layer of the architecture.

On the other hand, the deployment of RNs in WSNs provides an effective approach to address different optimization objectives. Among of which are those targeting better connectivity, full coverage, limited energy consumption, improved performance, and effective cost management. Therefore, the optimization of the RN placement in WSNs was approached considering single optimization objectives such as maximizing network lifetime [19-21] but mostly achieved with varying combinations of different objectives. The majority of the reviewed literature focused on cost-effective deployment and the minimization of the total number of RNs in addition to other critical objectives. These include establishing full connectivity among the deployed nodes in order to maintain sustainable WSN communications [10, 11, 22-30]. Minimizing overall energy consumption across the network is another objective that was also combined to maximize network lifetime [31-33]. Adopting network coverage as an integrated optimization objective was also considered to ensure more effective WSN deployment [34].

Other research efforts were also made to incorporate more objectives to address more effective optimization of RNPP. In some of these, network lifetime was improved with few RNs while maintaining fully connected networks [35-36] or ensuring better network coverage [37-38]. A solution targeting cost-effective deployment was also proposed to provide full network coverage and connectivity with the lowest number of RNs [11, 18, 39-40]. Other combinations considering the maximization of network lifetime, coverage, and connectivity without limitation on the number of deployed RNs were also considered [41-42].

This work, to the best of our knowledge, is the first attempt that integrates a greedy algorithm with the GA in order to guide it during convergence. The greedy algorithm plays a vital role in the optimization process. First, it provides an estimated number

of RNs for GA so that it can apply fixed-size encoding of a chromosome. The actual number of RNs that will be included for the solution in each chromosome is controlled by a set of predefined rules. These rules are flexible and can be controlled by algorithmic parameters so that the GA can get out of local optima. The outcome of the greedy algorithm is used to limit the search space complexity of the GA as well as to spot the places where multiple RNs might be placed to reduce energy.

III. ASSUMPTIONS

We consider a two-tiered hierarchical WSN architecture as shown in Figure 1. At the lower tier, there are a number of stationary sensor nodes which are battery-powered and equipped with short-range wireless transceivers of limited data rate. These are small-sized sensor devices that operate at low-power and sleep once they become idle. It is assumed that the sensor nodes are pre-deployed in the field at random positions or strategic locations over a 2D Euclidean area having no obstacles. They are only deployed to periodically collect application-specific data. The collected data is transmitted in packets of certain size to the RNs at the upper tier of the architecture. The transmissions take place over a network path of single hop at a certain time interval with constant transmission power. Received data at the RN are forwarded to the Internet infrastructure through base stations. Ideal situation is assumed such that there is no collision, external interference, nor retransmission during data communication.

It is assumed that the energy of sensor nodes is adequate to sense and collect the required data. Such operations would incur less power consumption compared to data reception/transmission. Therefore, power cost due to reception/transmission is only considered. On the other hand, RNs are not responsible for data collection and have no sensing modules. RNs receive and transmit data whereas sensor nodes only transmit data without reception of any data.

Another important assumption is the use of the same transmission and reception range for all the sensor nodes and RNs. Devices such as Tmote, MicaZ, and TelosB motes can reach up to 125m in outdoor environments. On the other hand, base stations are assumed to have enough range to interconnect all the deployed RNs in the deployment area. Connectivity between a sensor node and a certain RN is assumed according to the Euclidean distance. They are considered connected as long as they have an Euclidean distance shorter than the minimum of their communication ranges.

RNs are deployed to form a network topology of single hop connectivity without specific topological constraints. Using the single hop connectivity model would be feasible to maintain better lifetime of the network as sensor nodes are typically power-constrained devices. As they can be positioned differently, each RN can interconnect a varying number of sensor nodes. Therefore, RNs can end up with different data traffic loads and consume different transmission power. However, each RN is mains-powered or energy-harvesting device, thus there should be no concerns in regards to energy depletion of the RNs.

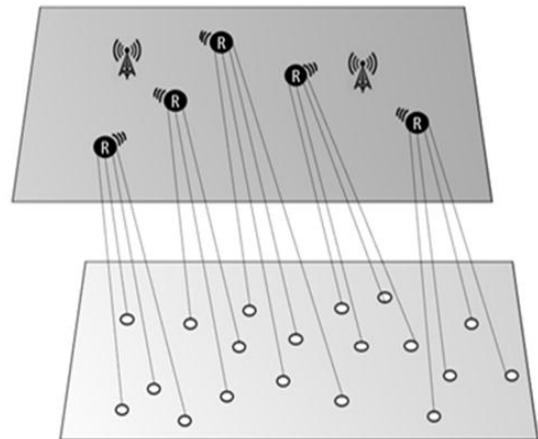


Fig. 1. Two-tiered Hierarchical WSN Architecture.

IV. MATHEMATICAL MODEL

The main objective of this work is to find out the applicability of integrating a greedy algorithm with a simple meta-heuristic optimization technique to solve large problem instances. We understand that there exist a number of powerful meta-heuristic algorithms in the literature and complex mathematical models to explain RNPP, but we focus on our primary objective. The technique of integrating a greedy algorithm with a simple meta-heuristic algorithm can be equally applicable to other powerful meta-heuristic algorithms. We did our experimentations on a very simple mathematical model to represent RNPP. We assume that a near optimal solution suggested by our solution to this simple model should be also a near-optimal solution to a relatively complex model. For the same reason, we further simplify RNPP by converting it to a single-objective optimization problem, where each of the objectives is multiplied with a weight.

The mathematical model of RNPP is formulated by first pre-processing the search spaces as described in subsection A. Accordingly, the locations of both sensor nodes and RNs are redefined, which is explained in subsection B. We then describe, in subsection C, the objective function considered for our RNPP. The notations used throughout this section are listed in Table 1.

IV.1. Search Space Pre-processing

Input to RNPP is a list of sensor nodes, each of which is described by a point. The list of points is drawn from a 2-D space. We will consider this as the search space associated with the given RNPP. Note that the unit of our search space is arbitrary. We limit this search space by computing both maximum and minimum values in both X and Y directions. The solution to RNPP is also a set of points, each of which describes the location of a RN. It should be noted that both input and output data are supposed to be continuous variables. This makes a RNPP computationally hard as the optimal points can have arbitrary precision during the optimization process. To overcome this situation, we build a mathematical model of our problem which deals with only integers while describing the locations of both sensor nodes and RNs. As we are looking for a near-optimal solution, this conversion does make sense [43].

Let d be a small value such that if a sensor or relay node's location is moved d units in X or Y direction, it causes negligible effect on our near-optimal solution, and the distance between any two sensor nodes is more than d . Let (x_{min}, y_{min}) , (x_{max}, y_{min}) , (x_{max}, y_{max}) , and (x_{min}, y_{max}) be the four corners of the smallest rectangle of our search space such that all input points (locations of sensor nodes) are inside of it and the distance from any sensor node to any side of the rectangle is more than d . For simplicity, let the length of each side of the rectangle be a multiple of d . We divide the rectangle by $((x_{max} - x_{min})/d - 1)$ straight lines that are parallel to Y and $((y_{max} - y_{min})/d - 1)$ straight lines that are parallel to X. These two groups of straight lines will divide the rectangle in such a way that it will have $((x_{max} - x_{min})/d) * ((y_{max} - y_{min})/d)$ number of squares, each of the squares has area d^2 units. This makes the discretization of any point in the search space straight forward, which can be described as follows: let (x_i, y_i) be a point in our search space, then this point is located in the square marked by a pair of two integers $(\text{floor}(x_i/d), \text{floor}(y_i/d))$.

The above procedure redefines not only the sensor nodes' positions but also the search space itself. Our new search space becomes a grid having $\text{ceil}((x_{max} - x_{min})/d)$ and $\text{ceil}((y_{max} - y_{min})/d)$ squares in X and Y directions, respectively. In this new search space, our computation involves only integers considering the locations of sensor nodes and RNs. The output to our integrated approach will be a set of pairs of integers. Each pair defines a particular square in the search space, into which we can place a RN. The actual position of the RN should not affect our near-optimal solution according to the definition of d .

Table 1. Units for Magnetic Properties

Symbol	Quantity
S	Sensor Nodes Matrix
s	Total Number of Sensor Nodes
Q	List of sensor nodes
d	The length of square
w	Physical Communication Range
c	Discretized Communication Range
R	RNs Matrix
r	Total Number of RNs
P	A list of pairs of integers that describes a solution to RNPP
max	Upper bound on the number of RNs
Rgreedy	RN Matrix from greedy algorithm
rgreedy	Total RNs by greedy algorithm
rfirstKind	Number of RNs of first kind
rsecondKind	Number of RNs of second kind
cgreedy	Discretized Communication Range used in greedy algorithm
I	Influence Matrix
couterCircle	Artificial Communication Range for first kind of RN in GA
mouterBox,	Extension of search space to have flexible
nouterBox	Number of RNs of second kind

IV.II. Redefinition of Point in Search Space Domain

Let $m = \text{ceil}((x_{max} - x_{min})/d)$ and $n = \text{ceil}((y_{max} - y_{min})/d)$. Now the sensor nodes can be described by S , a zero-one matrix of order m and n , where:

$$S[i][j] = \begin{cases} 1, & \text{if there is a sensor node in } (i, j) \text{ square,} \\ 0, & \text{Otherwise.} \end{cases}$$

s is the total number of sensor nodes such that $s = \sum_{0 < i < m, 0 < j < n} S[i][j]$.

In addition, Q is a list of s unique pairs of integers, where each pair (i, j) describes a square where a particular sensor is located or $S[i][j] = 1$.

Let w be a value such that a sensor node's signal can travel at best. Let $c = \text{floor}(w/d)$. It means that a sensor can communicate with a RN if and only if the distance between the RN and the sensor node is at most c square units. In other words, if the nearest RN of a sensor node is c squares away from it in both X or Y directions, the sensor node cannot communicate with the nearest RN.

R is a zero-one matrix of order m and n to store the positions of RNs, as follows:

$$R[i][j] = \begin{cases} 1, & \text{if there is a RN in } (i, j) \text{ square,} \\ 0, & \text{Otherwise.} \end{cases}$$

Let r be the total number of RNs, which we expect to find by our solution. Then, $r = \sum_{0 < i < m, 0 < j < n} R[i][j]$.

The output of our solution is P , a list of r unique pairs of integers, where each pair (i, j) describes a square in our search space in the 2-D space and $R[i][j] = 1$. We consider P as a near-optimal solution to RNPP.

IV.III. Objective Functions

An optimal solution to a RNPP is to find out the locations for a set of RNs, such that (i) the number of RNs in the set is minimum, (ii) each of the sensor nodes can reach at least one RN, and (iii) the energy consumed by each sensor node is minimized. We will call each of these expected properties as *Cost*, *Coverage* and *Energy*, respectively. We expressed each of these properties by a numerical value as follows:

(i) *Cost*: the cost associated with the procurement, installment, and maintenance of RNs that are suggested in a candidate solution. Let, max be an integer that we assume to be an upper bound for the expected number of RNs. So, for any candidate solution, $P_{candidate}$, let $r_{candidate}$ be the number of RNs to solve a RNPP, where $r_{candidate} \leq max$. We compute cost as the ratio between $r_{candidate}$ and max as follows:

$$f_1 = \frac{r_{candidate}}{max} \quad (1)$$

This ratio is then multiplied by 100 to normalize it with other metrics. This normalized value is further multiplied with a weight factor $w1$.

(ii) *Coverage*: the fraction of sensor nodes that can communicate with at least one RN. This value is further multiplied with a weight factor $w2$. Let s' be the number of

sensor nodes that can communicate with at least any RN for a given solution. Then, the network coverage rate is computed as follows:

$$f_2 = \frac{s'}{s} \quad (2)$$

(iii) *Energy*: Let the maximum distance possible for a sensor node to reach the nearest RN is limited by c squares. Let the sum of distances in square units by the sensors, which has the nearest RN in c square, be *Distance*. We compute energy consumption rate as the ratio between *Distance* and $s*c$ as follows:

$$f_3 = \frac{\text{Distance}}{s*c} \quad (3)$$

This ratio is then multiplied by 100 to normalize it. This normalized value is further multiplied with a weight factor w_3 .

The objective value for $P_{candidate}$ is the weighted sum of (1), (2), and (3) as follows:

$$F = w_1 f_1 + w_2 (-f_2) + w_3 f_3 \quad (4)$$

Accordingly, the optimization problem is formulated as a minimization problem. The actual values for the weights w_1 , w_2 and w_3 depend on user requirements. For example, we keep $w_1 < w_2$, if we emphasize more on coverage than cost. Algorithm 1 describes the above procedure.

Algorithm 1 OBJECTIVE

Input: Pcandidate, w1, w2, w3, rcandidate

Output: a real value that is the evaluation of the given candidate solution Pcandidate

Begin

```

1: total_cost = cost* rcandidate
2: norm_cost= w1*100*(total_cost/(max*cost))
3: covered_sensor = 0
4: total_distance = 0
5: total_dist_covered = 0
6: for each pair (i, j) in Q do
7:   (a, b) = closest(Pcandidate , (i, j))
      //closest() finds RN at (a,b) from Pcandidate
      //which is closest to (j,j)
8:   total_distance += distance (a, b) to (i, j)
9:   if distance (a, b) to (i, j) < c then
10:     ++covered_sensor
11:     total_dist_covered += distance (a, b) to (i, j)
12:   end if
13: end for
14: norm_energy= w3*100*(total_distance/(s*c))
15: norm_coverage= w2*100*(covered_sensor/s)
16: return (norm_energy + norm_cost +norm_coverage)
    
```

End

V. SOLUTION

Our greedy-integrated GA solution has two main steps. In the first step, we apply a greedy algorithm to find out some squares on the search space in which we observe signals from a maximum number of sensor nodes. In other words, the greedy algorithm tries to place RNs to reduce the energy consumption by sensor nodes. As the greedy algorithm is used to accelerate the convergence in the GA, we are not looking for any complete solution from it. Once the greedy algorithm returns the list of squares, in which RNs are in need, we instrument and execute the GA with the help of that list of squares. The following subsections provide detailed descriptions of these two stages: the greedy algorithm and greedy-integrated GA.

Algorithm 2 ComputingInfluence

Input: Set G

Output: Matrix I of order m and n

Begin

```

1: Create a zero matrix  $I$  of order  $m$  and  $n$ 
2: for each sensor  $g$  in  $G$ ; where  $g$  is at  $(i, j)$  do
3:   for each square  $(x, y)$  in  $I$  do
4:     if distance from  $(i, j)$  to  $(x, y) \leq C_{greedy}$  then
5:        $I[x][y] = I[x][y] + 1$ 
6:     end if
7:   end for
8: end for
9: return  $I$ 
End
    
```

Algorithm 3 GREEDY

Input: Set Q

Output: R_{greedy}

Begin

```

1: Set  $Q_I = Q$ 
2: Create a zero matrix  $R_{greedy}$  of order  $m$  and  $n$ 
3: While  $Q_I \neq \text{null}$  do
4:    $I = \text{ComputingInfluence}(Q_I)$ 
5:    $I(i,j) = \max(I)$ 
6:    $R_{greedy}[i][j] = 1$ 
7:   for each sensor  $g$  in  $Q_I$ ; where  $g$  is at  $(x, y)$  do
8:     if distance from  $(x, y)$  to  $(i, j) \leq C_{greedy}$  then
9:       delete  $g$  from  $Q_I$ 
10:    end if
11:  end for
12: end while
13: return  $R_{greedy}$ 
End
    
```

V.I. The Greedy Algorithm

We need to list the following notations in order to explain the working principle of our greedy algorithm. Let R_{greedy} be a zero-one matrix of order m and n , where:

$$R_{greedy}[i][j] = \begin{cases} 1, & \text{if GREEDY puts a RN in square } (i, j), \\ 0, & \text{Otherwise.} \end{cases}$$

Let r_{greedy} be the total number of RNs that our greedy algorithm suggests to be placed across the search space. It is clear that $r_{greedy} = \sum_{0 < i < m, 0 < j < n} R_{greedy}[i][j]$.

In the greedy algorithm, we assume that each sensor node can connect to a RN if the distance between them is equal or smaller than c_{greedy} squares. c_{greedy} is an artificial value such that $c_{greedy} > c$. It is expected that r_{greedy} is inversely proportional to the value of c_{greedy} .

Our greedy algorithm is explained with the ComputingInfluence and GREEDY algorithm as presented in Algorithm 2 and 3, respectively. The input to ComputingInfluence is a set of sensor nodes, G . The output of this algorithm is I , an integer matrix of order m and n . $I[i][j]$ is the number of sensor nodes from G whose signal can be reached at square (i, j) assuming that the signal from each sensor node can be propagated c_{greedy} squares in all directions.

The GREEDY algorithm is an iterative algorithm. The input to this algorithm is Q , which is the set of all sensor nodes. In each iteration, it computes I using ComputingInfluence. It then finds the maximum value in I . Let $I[i][j]$ contains maximum value, it then updates with $R[i][j] = 1$ and deletes the sensor nodes which could be served by placing a RN at (i, j) before starting the next iteration (if any).

V.II. The Greedy-Integrated Algorithm

The GA algorithm is executed once the greedy algorithm has completed its execution. Two very important pieces of information are obtained from the greedy algorithm: (i) the approximate number of RNs that can be used to serve all sensor nodes, (ii) the squares in the search space where RNs are more likely to be placed. This two outputs from the greedy algorithm will be used to effectively configure a GA to solve the RNPP. Hence, we call it an integrated approach.

There are two kinds of RNs that are expected from the GA algorithm. These are:

(i) $r_{firstKind}$: First Kind RNs are the relay nodes that are being tried to place near to the squares suggested by the greedy algorithm. To describe this kind of RNs, we need to introduce another artificial communication range $c_{outerCircle}$, where $c_{greedy} > c_{outerCircle} > c$. In total, GA will try to place r_{greedy} number of RNs of this kind. The range of each of the RN's coordinate values is by both adding and subtracting c_{greedy} with the corresponding coordinate positions from greedy algorithm. The main purpose of these RNs is to increase the coverage metric. During the convergence phase of GA, a RN will be considered for fitness value if it is, by random choice, not more than $c_{outerCircle}$ away from the position suggested by the greedy algorithm. This technique gives us the flexibility for our GA on the number of RNs. We use $r_{firstKind}$, to represent the total number of RNs of first kind.

(ii) $r_{secondKind}$: Second Kind RNs are the RNs that are being tried to place anywhere in the search space. To describe this kind of RNs, we need to introduce $m_{outerBox}$, $n_{outerBox}$, where $m_{outerBox} >$

m and $n_{outerBox} > n$. The GA will try to place a number of RNs of second kind. The range of coordinates of each RN is defined as $(0, m_{outerBox})$ and $(0, n_{outerBox})$. The main purpose of these RNs is to get rid of local optima that might be caused by those of first kind. The number of RNs of second kind will depend on the complexity of RNPP. If the solution from the GA algorithm is not satisfactory considering energy consumption and network coverage, it can be increased. During the convergence phase of the GA, a RN will be considered for fitness value if it is, by random choice, placed in the legal boundary of the search space. This technique gives us the flexibility for our GA on the number of RNs. We use $r_{secondKind}$, to represent the total number of RNs of second kind.

VI. EVALUATION

This section provides a description of the experiments carried out to evaluate the proposed integrated approach. It also presents and discusses the obtained experimental results.

VI.I. Experimental Setup

To demonstrate its efficiency, the proposed greedy-integrated GA was compared with a baseline algorithm, namely the classical GA algorithm. Both algorithms were tested using an experimental dataset, consisting of four randomly generated instances to represent fairly complex RNPP scenarios of small to large setups. In each instance, the deployment of a static traditional WSN was considered with a square area of a different dimension and sensor node count. These are as listed in Table 2. The number of sensor nodes in each scenario was set to the minimum needed to cover the whole area. Figure 2 shows the case of SC3 which presents a total of 250 sensor nodes deployed in a square region of 500×500 m² in size with a random distribution. The communication ranges of the sensor nodes and RNs in all the scenarios were assumed to be 40m [44, 45] and d was set to 4m.

Both algorithms were implemented using MATLAB R2018b, and executed using a system having an Intel Core i5, 2.5 GHz CPU, 4 GB RAM, and Mac OS. The basic parameters of the GA were similarly set for both algorithms. The initial population varies for the different scenarios and increases as the size of the network increases. For both models, the number of RNs is given by the greedy algorithm as an input to the GA process. However, the proposed greedy-integrated GA, the number of decision variables or the number of RNs that the GA will try to place is the sum of RNs of first, and second kinds. The domain of each of the RNs of first kind is defined by the corresponding RN from greedy algorithm and $c_{outerCircle}$. The domain of each of the RNs of the second kind is defined as $(0, m_{outerBox})$ and $(0, n_{outerBox})$. The crossover rate was configured to be 8%. In regards to the stop condition, the stall generation was set to 50 and the function tolerance value was $1e^{-3}$. The values of the weight factors, W_1 , W_2 and W_3 , can be any nonzero positive value. For example, we keep $W_1 > W_2 > W_3$ if cost is the most important objective over coverage and energy and coverage is more important than energy.

Table 2 . Basic Details of the Experimental Instances

SCENARIO	AREA SIZE (M2)	<i>s</i>	<i>R</i>
SC1	200×200	100	10
SC2	300×300	150	17
SC3	500×500	250	40
SC4	800×800	400	81

To account for randomness in the GA procedure, each algorithm was run for 30 independent times on each instance. The confidence level was set to 95%. The obtained results were then averaged over the 30 runs.

The performance of the proposed integrated approach was evaluated by establishing a comparison with the performance of the classical GA. The comparison was carried out using two main evaluation measures, similar to [16, 17, 40]. First is the solution quality indicated by the objectives values. This was collected for the objectives: network coverage, energy consumption, and cost. The second measure was time efficiency for which we considered a machine-independent metric. That is the total number of generations run by the algorithm before the code terminates.

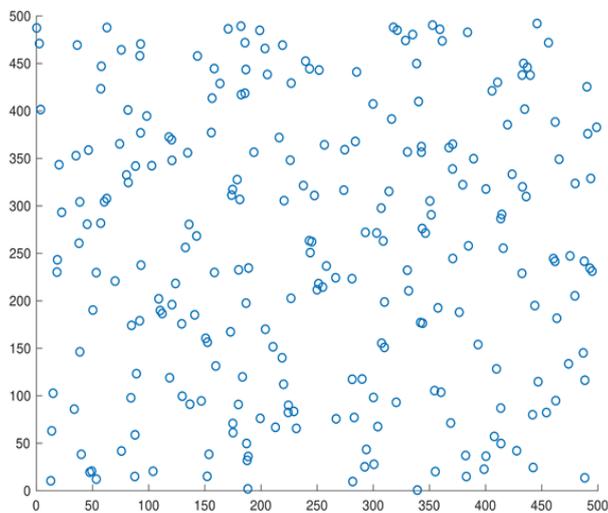


Fig 2. Representation of the SC3 dataset

VI.II. Results and Discussion

Tables 3 and 4 summarize the average values of the considered objective functions for all the instances. It is evident that the proposed integrated algorithm performed similar to the classical GA in SC1 with a relatively small setup. Figure 3 also shows that the same number of RNs was outputted by both algorithms. However, the greedy-integrated GA algorithm converged to the near-optimal solution with less GA iterations. As it can be seen in Figure 4, it required about 32% less generations before reaching terminating condition.

Increasing the size of the experimental setup resulted in overall degradation of the performance of both algorithms as indicated by the obtained results. Nevertheless, less performance degradation was experienced by the proposed approach in these

scenarios. Comparing the results of the classical GA in SC2 to those of SC3, the network coverage rate decreased by about 19% and the power consumption rate increased by about 14%. For the same scenarios, the proposed algorithm experienced less reduction on the network coverage rate (decreased by only 8%) and a less increase on the power consumption rate (increased by only 7%).

Table 3. Network Coverage Results

SCENARIO	CLASSICAL-GA		GREEDY-INTEGRATED GA	
	MEAN	STD	MEAN	STD
SC1	99.5	±0.9	100	±0
SC2	90.5	±2.8	95.2	±2.1
SC3	72.9	±2.7	86.8	±0.8
SC4	60.1	±2.7	77.4	±0.5

Table 4. Energy Consumption Results

SCENARIO	CLASSICAL-GA		GREEDY-INTEGRATED GA	
	MEAN	STD	MEAN	STD
SC1	60.6	±1.5	60.1	±1.7
SC2	66.5	±2.1	65.4	±1.6
SC3	75.5	±1.4	70.1	±1.1
SC4	81.7	±1.3	74.9	±1.2

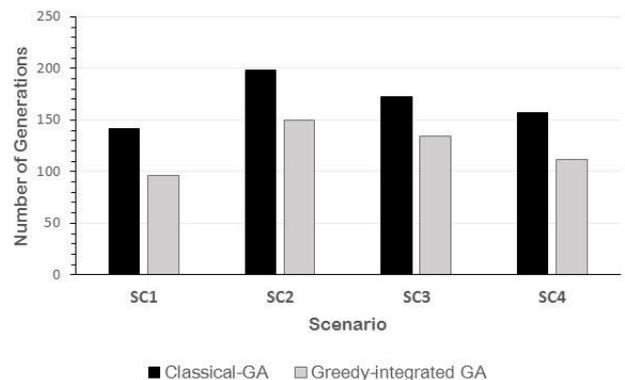


Fig 3. Number of Generation Comparison.

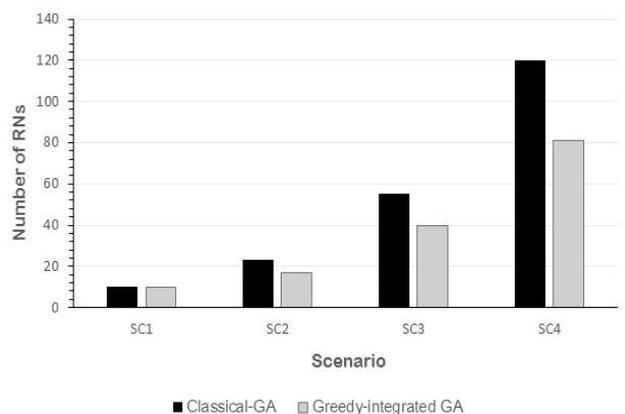


Fig 4. Number of RNs Comparison.

As the setup size increased, better improvement on the considered objective was achieved by the proposed algorithm compared to the classical GA. For example, it succeeded in Scenario SC3 to provide an increase by about 19% on the network coverage rate and a decrease by about 7% on the power consumption rate. Figure 5 and 6 show how the RNs were positioned by the classic GA and the proposed integrated algorithm, respectively. This is for the SC3 case presented in Figure 2 and after having the search space pre-processed. In a relatively very large setup such as SC4, the rates of network coverage and power consumption improved by about 29% and 8%, respectively, compared to the classical GA. Figure 3 also shows that the proposed algorithm required less number of RNs to perform similar to GA. It succeeded to reduce the RNs number by more than 25% particularly in large scenarios.

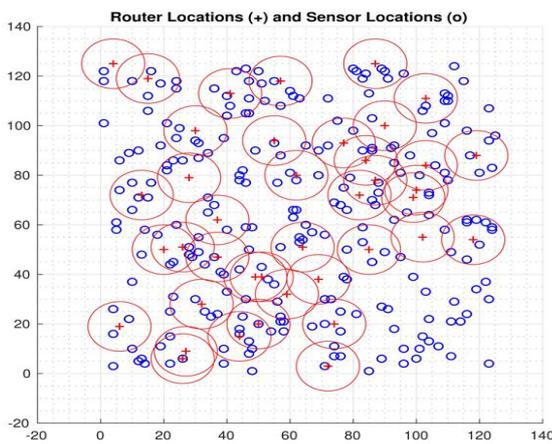


Fig 5. Output of the Classical GA for Scenario SC3.

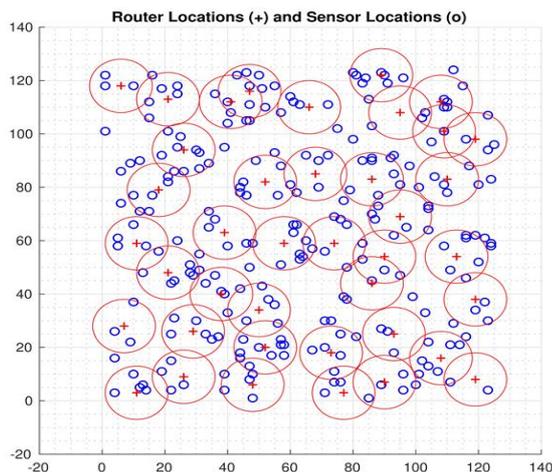


Fig 6. Output of the Proposed Algorithm for Scenario SC3.

It can also be noticed from Figure 4 that the proposed algorithm tended to converge faster than the classical GA in small and large setups. In Scenario SC4, for example, the algorithm required a number of generations less than the classical GA by about 29% before reaching terminating condition. Overall improvement of more than 22% on the convergence rate was achieved by our algorithm in all the instances. It is important to note that there is a randomness effect which needs to be considered in this regard. However, it was observed that the proposed algorithm managed to minimize such effect to some extent.

In further experiments, we compared our proposed integrated approach with the classical GA when keeping the generation number fixed at the value of 1000. In other words, we want to investigate the convergence or improvement achieved by each algorithm if it is given more number of generations. We only considered the large scenarios, SC3 and SC4, as we found that the GA did not perform well on these two large problems. One might think that the GA might perform well for these cases, if we keep it running for more generations. We performed this experiment to check this statement experimentally. In the case of scenario SC3, we compared the convergence of both algorithms when similar performance in terms of network coverage and power consumption was achieved. For scenario SC4, we compared the solution quality of both algorithms after reaching the maximum number of generations. In both setups, the experiments were carried out with varying numbers of RNs for further performance investigation.

Table 5 presents the results of both algorithms in the case of Scenario SC3. It is evident that less number of generations is required by the greedy-integrated GA to achieve similar performance of the classical GA. The integrated algorithm effectively succeeded in converging to a better solution with 33% less GA-iterations for 40 RNs. It can also be noticed that the integrated algorithm achieved almost 100% network coverage rate with 46 RNs and 15% less number of generations. For the other case, the results of both algorithms considering Scenario SC4 is presented in Table 6. It can be seen that the proposed algorithm outputted better solution quality compared to the classical GA. For network coverage, the proposed algorithm succeeded in covering about 90% of the sensor nodes with 97 RNs whereas the classical GA at most achieved coverage rate of less than 70%. It is also evident that the proposed algorithm reduced the rate of energy consumption by about 8% in average considering the deployment of both 81 and 97 RNs.

VII. CONCLUSION

The novel integrated approach introduced in this paper addresses the complexity of the search space considering unconstrained placement of RNs in large-scale WSNs. An effective greedy placement strategy is integrated with the GA to effectively support its search process and accelerate its convergence. The experimental results demonstrated how the proposed approach succeeded towards such a goal and achieved a considerable reduction in the GA iterations. Moreover, it was found that the classical GA did not perform well in large WSN deployments, and that the greedy technique helped it to effectively output better solution quality. The proposed greedy-integrated GA was able to increase network coverage and reduce power consumption while providing cost-effective solutions saving at most one third of RNs. This is evident even after varying the configurations of the GA by increasing and fixing the maximum number of generations and adding more populations. Such improvements provide an effective contribution towards better WSN deployments for different applications with large-scale setups. However, no noticeable improvement was achieved in small-scale scenarios. In future, we plan to extend our work to other variations of the RNPP considering varying-scale scenarios. The focus will be on establishing k-connectivity in multihop and 3D setups for heterogeneous WSN deployments.

TABLE 5 . CONVERGENCE RATE AND SOLUTION QUALITY FOR SCENARIO SC3

	Classical-GA						Greedy-integrated GA					
	RNs= 40			RNs= 46			RNs= 40			RNs= 46		
	Mean	Best	STD	Mean	Best	STD	Mean	Best	STD	Mean	Best	STD
Coverage Rate	91.14	92.80	±1.3	92.48	93.60	±1.6	95.76	97.20	±1.2	97.68	98.4	±1.0
Energy Cons. Rate	69.44	68.22	±1.0	66.84	64.75	±1.2	67.56	63.54	±2.4	62.61	61.75	±0.7
No of Generations	1000	1000	-	1000	1000	-	675	350	-	850	550	-

TABLE 6. CONVERGENCE RATE AND SOLUTION QUALITY FOR SCENARIO SC4

	Classical-GA						Greedy-integrated GA					
	RNs= 81			RNs= 97			RNs= 81			RNs= 97		
	Mean	Best	STD	Mean	Best	STD	Mean	Best	STD	Mean	Best	STD
Coverage Rate	67.05	69.25	±2.0	71.70	73.50	±1.7	83.15	84.5	±1.2	89.05	90.5	±1.2
Energy Cons. Rate	79.76	78.55	±0.8	77.21	74.92	±1.6	73.80	72.48	±1.2	69.50	68.85	±0.4
No of Generations	1000	1000	-	1000	1000	-	1000	1000	-	1000	1000	-

REFERENCES

[1] P. Parwekar, and S. Rodda, "Optimization of Clustering in Wireless Sensor Networks Using Genetic Algorithm," *Int. J. Appl. Metaheuristic Comput.*, vol. 8, no. 4, pp. 84-98, 2017.

[2] P. S. Ragavan, and K. Ramasamy, "Optimized routing in wireless sensor networks by establishing dynamic topologies based on genetic algorithm," *Cluster Comput.*, vol. 22, no. 5, pp. 12119-12125, 2019.

[3] V. Sivakumar, and D. Rekha, "Node scheduling problem in underwater acoustic sensor network using genetic algorithm," *Pers. Ubiquitous Comput.*, vol. 22, no. 5-6, pp. 951-959, 2018.

[4] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 550-586, 2016.

[5] D. S. Deif, and Y. Gadallah, "Classification of wireless sensor networks deployment techniques," *IEEE Commun. Surveys Tuts.*, vol. 16, no.2, pp. 834-855, 2014.

[6] M. Iqbal, M. Naeem, A. Anpalagan, N. N. Qadri, and M. Imran, "Multi-objective optimization in sensor networks: Optimization classification, applications and solution approaches," *Comput. Networks.*, vol. 99, pp. 134-161, 2016.

[7] J. M. Lanza-Gutiérrez, N. Caballé, J. A. Gómez-Pulido, B. Crawford, and R. Soto, "Toward Robust Multi-Objective Metaheuristic for Solving the Relay Node Placement Problem in Wireless Sensor Networks," *IEEE Sensors J.*, vol. 19, no. 3, p. 677, 2019.

[8] J. M. Lanza-Gutierrez, and J. A. Gomez-Pulido, "Assuming multiobjective metaheuristics to solve a three-objective optimization problem for relay node deployment in wireless sensor networks," *Appl. Soft Comput.*, vol. 30, pp. 675-687, 2015.

[9] B. O. Ayinde, and H. A. Hashim, "Energy-efficient deployment of relay nodes in wireless sensor networks using evolutionary techniques," *Int. J. Wirel. Inf. Networks.*, vol. 25, no. 2, pp. 157-172, 2018.

- [10] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm for k-connected relay node placement in wireless sensor networks," in Proc. of the second international conference on computer and communication technologies, Springer, New Delhi, 2016, pp. 721-729.
- [11] S. K. Gupta, P. Kuila, and P. K. Jana, "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks," *Comput. Electr. Eng.*, vol. 56, pp. 544-556, 2016.
- [12] J. George, and R. M. Sharma, "Relay node placement in wireless sensor networks using modified genetic algorithm," in 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology, 2016, pp. 551-556.
- [13] S. Misra, N. E. Majd, and H. Huang, "Constrained relay node placement in energy harvesting wireless sensor networks," in IEEE 8th International Conference on Mobile Ad-Hoc and Sensor Systems. 2011, pp. 25-34.
- [14] N. E. Majd, "Toward Optimal Constrained Relay Node Placement Problem in Energy Cooperative Energy Harvesting Wireless Sensor Networks. *Int. J. Comput. Sci. Inf. Secur.*," vol. 15, no. 7, pp. 292-300, 2017.
- [15] V. Snášel, and L. P. Kong, "Modified Genetic Algorithm with Sorting Process for Wireless Sensor Network," in International Conference on Innovative Computing and Communications, Springer, Singapore, 2020, pp. 381-388.
- [16] E. F. Flushing, and G. A. Di Caro, "Exploiting synergies between exact and heuristic methods in optimization: an application to the relay placement problem in wireless sensor networks," in International Conference on Bio-Inspired Models of Network, Information, and Computing Systems, Springer, Cham, 2012, pp. 250-265.
- [17] Y. H. Zhang, Y. J. Gong, T. L. Gu, Y. Li, and J. Zhang, "Flexible genetic algorithm: A simple and generic approach to node placement problems," *Appl. Soft Comput.*, vol. 52, pp. 457-470, 2017.
- [18] G. V. Singh, S. Harizan, and P. Kuila, "Quantum Inspired Genetic Algorithm for Relay Node Placement in Cluster Based Wireless Sensor Networks," in International Conference on Computational Intelligence, Communications, and Business Analytics, Springer, Singapore, 2018, pp. 381-391.
- [19] N. T. Tam, H. T. T. Binh, T. H. Hung, and D. A. Dung, "Prolong the Network Lifetime of Wireless Underground Sensor Networks by Optimal Relay Node Placement," in International Conference on the Applications of Evolutionary Computation, Springer, Cham, 2019, pp. 439-453.
- [20] T. N. Mathaba, "Optimal Sink-Node Placement and Routing for an Energy Efficient Two-Tier Wireless Sensor Network," in 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems, 2018, pp. 1-6.
- [21] C. Ma, W. Liang, and M. Zheng, "Lifetime Constrained Relay Node Placement in WSNs: A Cluster-Based Approximation Algorithm," in 2017 IEEE 85th Vehicular Technology Conference, 2017, pp. 1-5.
- [22] F. Senel, and M. Younis, "Novel relay node placement algorithms for establishing connected topologies," *J. Netw. Comput. Appl.*, vol. 70, pp. 114-130, 2016.
- [23] S. Sapre, and S. Mini, "Optimized relay nodes positioning to achieve full connectivity in wireless sensor networks," *Wirel. Pers. Commun.*, vol. 99, no. 4, pp. 1521-1540, 2016.
- [24] Y. Li, C.S. Chen, K. Chi, and J. Zhang, "Two-tiered relay node placement for WSN-based home health monitoring system," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 3, pp. 589-603, 2019.
- [25] C. Ma, W. Liang, M. Zheng, and H. Sharif, "A connectivity-aware approximation algorithm for relay node placement in wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 2, pp. 515-528, 2015.
- [26] C.C Chen, C. Y. Chang, and P.Y. Chen, "Linear time approximation algorithms for the relay node placement problem in wireless sensor networks with hexagon tessellation," *J. Sensors.*, 2015.
- [27] G. Liu, K. Lu, and J. Li, "Approximation Algorithm for Relay Node Placement in Singled-Tiered Wireless Sensor Networks," in 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics, 2019, pp. 162-167.
- [28] K. Tisseli, C. Benzaid, N. Lasla, and N. Badache, "Connectivity-aware Relay Node Deployment in Grid-based Wireless Sensor Networks," in 2019 Wireless Days, 2019, pp. 1-4.
- [29] S. Sapre, and S. Mini, "Relay node placement to establish full connectivity in directional sensor networks," in 2017 Fourteenth International Conference on Wireless and Optical Communications Networks, 2017, pp. 1-5.
- [30] S. Sapre, and S. Mini, "Moth Flame Based Optimized Placement of Relay Nodes for Fault Tolerant Wireless Sensor Networks," in 2018 9th International Conference on Computing, Communication and Networking Technologies, 2018, pp. 1-6.
- [31] B. Yuan, H. Chen, and X. Yao, "Optimal relay placement for lifetime maximization in wireless underground sensor networks," *Inf. Sci. (Ny).*, vol. 418, pp. 463-479, 2017.
- [32] B. Yu, W. Yuanping, Z. Liang, H. Yuan, and Z. Aijuan, "Relay node deployment for wireless sensor networks based on PSO," in 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 2393-2398.

- [33] O. Ozkan, and M. Ermis, "Nature-inspired relay node placement heuristics for wireless sensor networks," *J. Intell. Fuzzy Syst.*, vol. 28, no. 6, pp. 2801-2809, 2015.
- [34] C. Ma, W. Liang, M. Zheng, and H. Sharif, "A novel local search approximation algorithm for relay node placement in wireless sensor networks," in *2015 IEEE Wireless Communications and Networking Conference*, 2015, pp. 1518-1523.
- [35] D. Djenouri, and M. Bagaa, "Energy-aware constrained relay node deployment for sustainable wireless sensor networks," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 1, pp. 30-42, 2017.
- [36] B. O. Ayinde, and H. A. Hashim, "Energy-efficient deployment of relay nodes in wireless sensor networks using evolutionary techniques," *Int. J. Wirel. Inf. Networks.*, vol. 25, no. 2, pp. 157-172, 2018.
- [37] J. M. Lanza- Gutierrez, and J. A. Gomez- Pulido, "A gravitational search algorithm for solving the relay node placement problem in wireless sensor networks," *Int. J. Commun. Syst.*, vol. 30, no. 2, 2017.
- [38] D. Djenouri, and M. Bagaa, "Energy harvesting aware relay node addition for power-efficient coverage in wireless sensor networks," in *2015 IEEE International Conference on Communications*, 2015, pp. 86-91.
- [39] K. Nitesh, and P. K. Jana, "Relay node placement with assured coverage and connectivity: a Jarvis March approach," *Wirel. Pers. Commun.*, vol. 98, no. 1, pp. 1361-1381, 2018.
- [40] M. A. Benatia, M. H. Sahnoun, D. Baudry, A. Louis, A. El-Hami, and B. Mazari, "Multi-objective WSN deployment using genetic algorithms under cost, coverage, and connectivity constraints," *Wirel. Pers. Commun.*, vol. 94, no. 4, pp. 2739-2768, 2017.
- [41] H. Zhang, Z. Zhang, F. Zhang, L. Li, and Y. Wang, "Optimized design of relay node placement for industrial wireless network," *Int. J. Distrib. Sens. Networks.*, vol. 10, no. 11, p. 759826, 2014.
- [42] J. M. Lanza-Gutiérrez, J. A. Pulido, and M. A. Vega-Rodriguez, "A New Realistic Approach for the Relay Node Placement Problem in Wireless Sensor Networks by Means of Evolutionary Computation," *Ad Hoc Sens. Wirel. Networks.*, vol. 26, no. 1-4, pp. 193-209, 2015.
- [43] Y. Yoon and Y. Kim, "An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1473-1483, Oct. 2013, doi: 10.1109/TCYB.2013.2250955.
- [44] J. T. Adams, "An introduction to IEEE STD 802.15.4," in *2006 IEEE Aerospace Conference, Big Sky, MT, 2006*, p. 8, doi: 10.1109/AERO.2006.1655947.
- [45] S. C. Ergen, "Zigbee/IEEE 802.15. 4 summary," *UC Berkeley*, September, vol. 10, p. 17, 2004.