Genetic Algorithm for *k*-Connected Relay Node Placement in Wireless Sensor Networks

Suneet K. Gupta, Pratyay Kuila and Prasanta K. Jana

Abstract Wireless Sensor Networks (WSNs) are widely used for many applications including health care, environment monitoring, underground mines, and so on. In WSN, deployment of relay nodes to cover specific region or target is an important issue. In a target-based WSN, it is important that all the targets must be covered by sensor nodes, and the sensor nodes are connected with the backbone network. In this paper, we propose two algorithms for relay node placement which provide *k*-connectivity of the sensor nodes. The first algorithm is based on Genetic Algorithm (GA), and the second one is based on greedy approach. We have also to extensively simulate both the algorithms to study their performance.

Keywords Potential positions · Relay node · *k*-connected · Wireless sensor networks · Genetic algorithm

1 Introduction

The design of wireless sensor networks (WSNs) is a complicated task with substantial impact on its efficiency design and cost in real-life applications. Sensor nodes are tiny electronic devices with limited memory, energy, and transmitting capabilities [1]. An important goal of this type of network is to prolong the life time and make highly fault tolerant so that the sensor nodes cover a region of interest and forward the important data to the remote base station (BS) directly or by multi-hop

S.K. Gupta (✉)
Department of Computer Science and Engineering, OPJIT, Raigarh 496001, India
e-mail: suneet.banda@gmail.com

P. Kuila
School of Computer Engineering, KIIT University, Bhubaneswar 751024, India
e-mail: pratyay_kuila@yahoo.com

P.K. Jana
Department of Computer Science and Engineering, ISM, Dhanbad 826004, India
e-mail: prasantajana@yahoo.com

S.C. Satapathy et al. (eds.), *Proceedings of the Second International Conference on Computer and Communication Technologies*, Advances in Intelligent Systems and Computing 379, DOI 10.1007/978-81-322-2517-1_69

communication. Connectivity among the sensor nodes is crucial for most of the applications because the possible partition of the network may cause undesirable effects on the amount of information forwarded to the interested users [2]. Connectivity defines that individual sensor nodes are connected with different relay nodes to forward its data to the BS, and k-connectivity indicates that each sensor node is connected with at least k different relay nodes. Here, our objective is to design an efficient algorithm to place minimum number of relay nodes to provide desired k-connectivity. Note that the above k-connectivity problem is NP-hard [2]. Therefore, GA-based algorithm or a greedy-based heuristic is suitable for finding the approximate solution.

In [3], authors have surveyed various strategies of node deployment to achieve coverage and connectivity. Meguerdichian et al. [4] have pointed out that the coverage objective is a measure of the Quality of Service (QoS) that is provided by a particular network design. Another solution to maximize network lifetime is to deploy redundant sensor nodes in an area and periodically use active and inactive nodes. Martins et al. [5] suggest to periodically re-arrange active and inactive nodes to prolong the network lifetime. The most popular proactive strategy for preserving the network connectivity in the presence of a faulty node is to carefully place redundant sensor nodes during or after the initial deployment of a WSN. The sensor placement can be grouped into two categories. The first tries to just establish connectivity between end points, i.e., $k = 1$ [6]. In the second category, higher degrees of connectivity are to be achieved, i.e., $k > 1$ [7]. Given the scope of this section, we focus on the second category.

In this paper, we have presented LPP formulation of k-connected relay node placement problem and two algorithms: one is based on GA, and another is greedy approach. We describe all the basic steps of GA, i.e., initial population, selection, crossover, and mutation for the solution of the problem. We perform the extensive simulation of our proposed algorithm and compare the results.

The rest of the paper is organized as follows. The terminologies and proposed work are described in Sect. 2. Experimental results are presented in Sect. 3 followed by conclusion in Sect. 4.

2 Terminologies and Proposed Work

The problem is given x number of potential positions and n number of targets, we have to select minimum number of potential positions to place relay nodes, so that the targets will be k-connected (for some value of k). Here, it is assumed that a sensor node is placed adjacent to a target. Therefore, we can say that target and sensor node is a single entity. Some pre-defined potential positions are there where we can place relay node to provide connectivity to the sensor nodes so that the sensor nodes can easily send their data to the base station (BS). It is also assume that the relay nodes have enough communication range to directly or indirectly communicate with the BS. We use the following terminologies in the proposed algorithms:

(1) The set of target (sensor node) is denoted by $\tau = \{\tau_1, \tau_2, \ldots \tau_n\}$.
(2) The set of relay nodes is denoted by $\zeta = \{\zeta_1, \zeta_2, \ldots\zeta_m\}$.
(3) The set of pre-defined potential positions is denoted by $P = \{p_1, p_2, \ldots p_x\}$.
(4) R_C denotes the communication range of the sensor nodes.
(5) dist(τ_i, ζ_j) denotes the Euclidian distance between τ_i and ζ_j.
(6) ComSet(τ_i) denotes the set of potential positions those are within the range of τ_i. In other words,

$$\text{ComSet}(\tau_i) = \{p_j | \text{dist}(\tau_i, p_j) \leq R_C \text{ and } p_j \in P\}$$

Therefore, we have to place a relay ζ_r on any one of the potential positions from ComSet(τ_i) to ensure connectivity of τ_i.
(7) W denotes the set of relay nodes, which provides k-connectivity to all targets and ω_i denotes the set of relay nodes that those are providing k-connectivity to τ_i. In other words, $\omega_i = \{\zeta_r \mid \text{dist}(\tau_i, \zeta_r) \leq R_C, \forall r, 1 \leq r \leq m\}$ and

$$W = \bigcup_{i=1}^{n} \varpi_i$$

8) cov_set (ζ_j) denotes the set of targets (sensors) that those are within communication range of ζ_j. In other words,

$$\text{cov\_set}(\zeta_j) = \{\tau_i | \text{dist}(\tau_i, \zeta_j) \leq R_C \text{ and } \tau_i \in \tau\}$$

2.1 LP Formulation for k-Connected Node Deployment Problem

Now, we address the k-connectivity problem where our basic objective is to minimize the number of relay nodes. Let p_i be a Boolean variable such that

$$p_i = \begin{cases} 1, & \text{If a } relay \text{ node is placed on } p_i, \forall p_i \in P \\ 0, & \text{Otherwise} \end{cases} \tag{2.1}$$

So, the Linear Programming (LP) formulation of the k-connectivity problem can be formulized as follows:
Minimize $Z = |W|$
Subject to

$$\varpi_i \geq k, \forall i, 1 \leq i \leq n \tag{2.2}$$

$$p_i \leq 1, \forall i, 1 \leq i \leq x \tag{2.3}$$

The constraint (2.2) states that the every target is connected with at least k number of relay nodes, i.e., k-connected. The constraint (2.3) ensures that maximum one relay node can be placed on any potential position p_i, $\forall i$, $1 \leq i \leq x$.

2.2 k-*Connected Algorithm Using GA*

Now we present the GA-based [8, 9] approach for providing the k-connectivity to all targets with chromosome representation, initial population selection, crossover, and mutation in the subsequent subsections as follows.

2.2.1 Chromosome Representation

We represent the chromosome as a string of relay nodes. The length of each chromosome is kept same as the number of target points. In a chromosome, ith gene value is a set of relay nodes those are placed on ComSet(τ_i) to ensure connectivity.

Let us assume a scenario with 11 targets 20 potential positions. Table 1 represents the individual target and the potential positions within its range. Figure 1 shows a chromosome for $k = 2$. Here, the gene value at position 7 is {10, 12} which implies that the ζ_{10} and ζ_{12} are placed on p_{10} and p_{12}, respectively, and therefore, now τ_7 is connected with ζ_{10} and ζ_{12}. Similarly, τ_{10} and τ_{11} are connected with {ζ_{14}, ζ_{17}} and {ζ_{12}, ζ_{14}}, respectively.

2.2.2 Fitness Function

The fitness value of a chromosome represents its qualification on the basis to provide k-connectivity to all targets using minimum number of relay nodes. We

Table 1 Targets and relay nodes

Targets (τ_i)	ComSet (τ_i)
τ_1	{1, 2}
τ_2	{1, 2, 7, 8, 20}
τ_3	{3, 4, 5, 20}
τ_4	{4, 5, 6}
τ_5	{5, 6, 10, 11}
τ_6	{7, 9, 14, 17}
τ_7	{10, 11, 12}
τ_8	{7, 8, 16}
τ_9	{16, 17, 18, 19}
τ_{10}	{14, 15, 17, 18}
τ_{11}	{12, 13, 14, 15}

Target no.	1	2	3	4	5	6	7	8	9	10	11
Gene value (ω_i)	{1, 2}	{1, 2}	{3, 5}	{5, 6}	{10, 11}	{7, 14}	{10, 12}	{7, 8}	{17, 19}	{14, 17}	{12, 14}

Fig. 1 Chromosome generated from Table 1

build a fitness function to evaluate the individual chromosomes of the initial population. Our aim is to find a minimal set of relay nodes with the consideration that this set provides *k*-connectivity to all the targets. Therefore, the fitness function for proposed work is to minimize $Z = |W|$.

Now, let us calculate the fitness value of the chromosome as shown in Fig. 1.

$$W = \bigcup_{i=1}^{n} \varpi_i = \{\{1, 2\} \cup \{1, 2\} \cup \{3, 5\} \cup \{5, 6\} \cup \{10, 11\} \cup \{7, 14\} \cup \{10, 12\} \cup \{7, 8\}$$
$$\cup \{17, 19\} \cup \{14, 17\} \cup \{12, 14\}\} = \{1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 14, 17, 19\}$$

i.e, $Z = |W| = 13$

Therefore, with reference to chromosome (refer Fig. 1) only 13 out of 20 relay nodes are enough to provide 2-connectivity to all targets.

2.2.3 Selection

The selection process determines which of the chromosomes from the current population will mate (crossover) to create new chromosomes. For the selection process, we select some valid chromosomes with higher fitness value. The individuals with better fitness values have better chances of selection. There are several selection procedures, e.g., Tournament selection, Rank selection, Roulette-wheel, etc. In our work, we have used Roulette-wheel selection.

2.2.4 Crossover

The crossover operation takes place between two randomly selected chromosomes. To produce the new offspring from the selected parent chromosomes, we use 1-point crossover whereby a point is chosen at random, and the two parent chromosomes exchange their information after that point. The whole process is shown in Fig. 2.

2.2.5 Mutation

The mutation is applied at a selected gene rather than randomly selected gene. In mutation operation, that relay node is selected from a chromosome which has

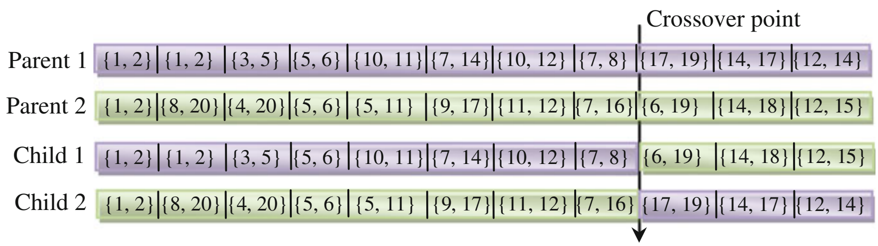

Fig. 2 Crossover operation

minimum appearance and tries to replace it with that relay node which has higher presence in the chromosome.

2.3 *Node Deployment Based on Greedy Approach*

In this method, the relay nodes are deployed for providing *k*-connectivity to all targets using greedy methods. For the same, there are two algorithms, e.g., Algorithm 1 and Algorithm 2, which are depicted in Figs. 3 and 4, respectively. In algorithm 1, we firstly generate cov_set(ζ_i).

This set stores all the targets, which is covered by ζ_i. After execution of Algorithm 1, Algorithm 2 will start its execution (refer Fig. 4). In Algorithm 2, output of Algorithm 1 acts as an input. The step 1 of Algorithm 2 sorts the ζ_i on |cov_set(ζ_i)|, i.e., number of targets covered by ζ_i. After then sub-step 2.1 selects that relay node ζ_i covers maximum number of targets. Sub-step 2.2 forms the resultant set of the relay nodes (*W*) which provide the *k*-connectivity to all targets.

Fig. 3 Algorithm for generating cov_set(ζ_i)

Algorithm 1: Formation of *cov_set* (ζ_j)

Input: Coordinates of targets and relay nodes
Output: Generation of *cov_set* (ζ_i), $\forall \zeta_j \in \zeta$
Procedure:
Step 1: /* Formation of *cov_set* (ζ_i), $\forall \zeta_j \in \zeta$ */
    for all $\zeta_j \in \zeta$
        for all $\tau_i \in \tau$
            if **(**$dist(\tau_i, \zeta_j) \le R_C$**)**
                1.1 **then** *cov_set* (ζ_j) = τ_i
            endif
        end for
    end for
Step 2: *Stop.*

Fig. 4 Greedy algorithm to find out W

Algorithm 2: *GreedyToFind*(*W*)

Input: $cov\_set\ (\zeta_i)$, $\forall i$, $1 \le i \le m$.
Output: Set of relay nodes which provide *k*-connectivity.
Procedure:
Initialization: $W = \{\}$
Step 1: Sort ζ_i, $\forall i$, $1 \le i \le m$ in according of $|cov\_set\ (\zeta_i)|$.
Step 2: ***While*** (τ_r $\forall r$, $1 \le r \le n$ is not *k*-connected)
  2.1: Select the ζ_i with highest $|cov\_set\ (\zeta_i)|$.
  2.2: $W = W \cup \{\zeta_i\}$
 End while
Step 3: Stop.

3 Experimental Results

We performed the experiments on the proposed algorithm using MATLAB R2012b. We depict the experimental results for the minimum number of relay nodes required for *k*-connectivity. For simulation purpose, we considered two different network scenarios (WSN #1 and WSN #2), and both of them have the sensing field of 100 × 100 and 300 × 300 square meter area, respectively. In both the scenarios, we have formed the grid of having size 10 m and position of relay nodes at intersecting points of these grids except boundary of the scenario. For execution of our proposed GA-based algorithm, we considered an initial population of 60 chromosomes. During selection operation of GA, we have selected 10% best population using Roulette-Wheel for performing 1-point crossover operation. We compare the simulated results of both algorithms.

Figure 5a, b shows the comparison of algorithms in terms of number of relay nodes required to provide k (≥2) connectivity in scenarios WSN #1 and WSN #2. With this figure, it is clearly shown that GA-based approach required less number of relay nodes to provide *k*-connectivity than greedy-based approach. GA-based proposed algorithm performed better than proposed Greedy-based approach because GA always gives the global optimal solution while greedy gives the local optimal solution. Moreover, in

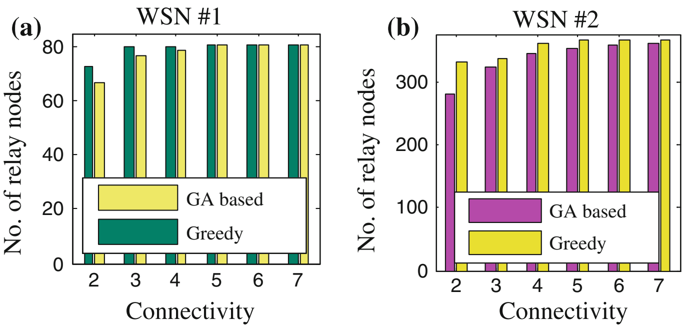

Fig. 5 Number of relay nodes in **a** WSN #1, **b** WSN #2

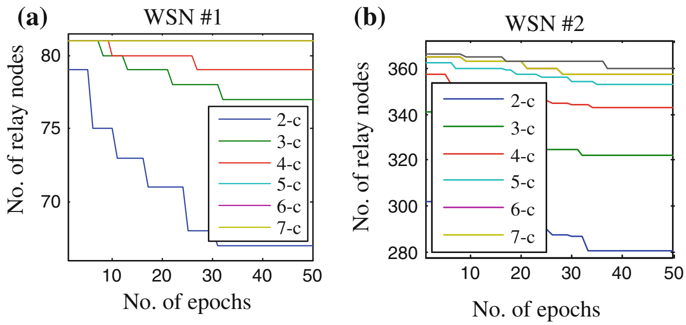

Fig. 6 Convergence of number of relay nodes in **a** WSN #1, **b** WSN #2

greedy, we make the local best choice on the basis of fitness function but in GA, choice is made with considering that in each iteration, it strengthens the value of fitness function. Figure 6a, b represents the convergence of fitness value of proposed algorithm in WSN #1 and WSN #2 scenarios, respectively. From the figure, it is clearly depicted that proposed algorithm starts giving the optimized value between 30 and 40 epochs. Both algorithms provide k-connectivity so these also provide the fault tolerance up to failure of $k - 1$ relay nodes complementary.

4 Conclusions

In this paper, we have proposed genetic algorithm-based node deployment algorithm for providing the k-connectivity to every targets (sensors). Here, we have given explanation for LPP formulation and step-by-step execution of all basic steps of GA with suitable examples. We have also proposed node deployment policy based on greedy approach. In the greedy approach, we arrange all the relay nodes based on number of connected targets, after then we start selecting the relay nodes. The algorithm stops execution till all the targets are not k-connected. From the experimental results, it is seen that GA-based approach is better than the greedy-based node deployment strategy. Our future research will be carried out to design energy aware relay node placement with generation of energy aware node disjoint sets.

References

1. Liu, X., Mohapatra, P.: On the deployment of wireless data back-haul networks. IEEE Trans. Wireless Commun. **6**, 1426–1435 (2007)
2. Konstantinidis, A., Yang, K., Zhang, Q., Zeinalipour-Yatzi, D.: A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. Comput. Netw. **54**, 960–976 (2010)

3. Younis, M., Akkaya, K.: Strategies and techniques for node placement in wireless sensor networks: a survey. Ad Hoc Network **6**, 621–655 (2007)
4. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. IEEE Infocom, pp. 1380–1387 (2001)
5. Martins, F.V.C., Nakamura, F.G., Quintao, F.P., Mateus, G.R.: Modeland algorithms for the density, coverage and connectivity control problem in flat WSNs. In: Proceedings of the International Network Optimization Conference (INOC'07), pp. 1145–1152 (2007)
6. Lloyd, E.L., Xue, G.: Relay node placement in wireless sensor networks. IEEE Trans. Comput. **56**(1), 134–138 (2007)
7. Nitesh, K., Jana, P.K.: Relay node placement algorithm in wireless sensor network. In: Proceedngs of IACC-2014, pp. 220–225. IEEE (2014)
8. Kuila, P., Gupta, S.K., Jana, P.K.: A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. Swarm Evol. Comput. **12**, 48–56 (2013)
9. Gupta, S.K., Kuila, P., Jana, P.K.: GAR: an energy efficient GA-based routing for wireless sensor networks. In: Proceedings of ICDCIT 2013, LNCS (Springer), vol. 7753, pp. 267–277 (2013)