Healing Partitioned Wireless Sensor Networks

Gaurav Kumar and Virender Ranga(✉)

Department of Computer Engineering,
National Institute of Technology, Kurukshetra, Haryana, India
gngauravnain@gmail.com, virender.ranga@nitkkr.ac.in

Abstract. Sensor nodes in wireless sensor networks are prone to failures because of their sensitiveness in the harsh surroundings. Sometimes, a failure of large scale nodes may occur in the deployed network and it converts connected network into disjoint segments called network partition problem. Therefore, a deployed application demands a continuous fault repairing mechanism to repair the lost connectivity. Deployment of additional relay node in damaged network is one of the best methods to restore the network operation. However, the relay node placement problem is shown to be an NP-hard problem. In this paper, we propose a swarm intelligence based solution to find the best locations of relay node placement in polynomial time. The simulation results show the performance gain of proposed solution over the state-of-the-art solutions.

Keywords: Connectivity · Relay node · Meta-heuristic · Swam intelligence

1 Introduction

In the recent years, a boost in the development of applications of wireless sensor networks (WSNs) has been observed. SNs have the power to sense, compute and transmit the data to other SNs or base station (BS) for further processing [1]. Sometimes, we deploy SN in such areas where recharge or replacement of the battery is impossible. Due to this type of applications, WSNs are prone to many types of failures such as hardware failure (unavailability of a power source, no coverage), natural disaster (volcano, flood, earthquake etc.) and explosion on the battlefield or coal mine. Therefore, it requires a special attention towards repair mechanisms for WSNs. A lot of work has been done to guarantee the normal functioning of all underlying operations. One of repairing mechanism is to connect different segmented parts of the disjoint network to provide inter-segment connectivity (partitions in WSN might be created due to large scale failures of SNs). This is a very well-known and well-researched problem of WSNs. A number of researchers find a different kind of solutions like (algorithmic based, approximation based, heuristic based) to restore lost connectivity in WSNs [2, 3]. With the help of additional SNs. However, SNs have smaller coverage and low connectivity. Therefore, we require a special type of node, i.e. relay node (RNs), to repair the disjoint partitioned network. RNs have larger battery power and longer coverage area compared to SNs. Relay node placement (RNP) requires advance knowledge about the communication range of SNs as well as RNs and working area of WSNs. The problem of placing RNs in the disjoint partitioned segments, to provide lost

S.F. Ochoa et al. (Eds.): UCAmI 2017, LNCS 10586, pp. 545–557, 2017.
DOI: 10.1007/978-3-319-67585-5_54

connectivity in WSN, is called relay node placement problem (RNPP) [3, 4]. Our main objective is to place a minimum number of relay nodes and maintain lost connectivity among all segments. Some solutions have been proposed to minimize the number of required RNs. Nowadays, a new era of meta-heuristic based solutions are suggested to solve NP-hard or NP-complete problems because of their simplicity, flexibility and capability to solve complex problem in polynomial time [5]. Problem independence and randomness are two main features of any meta-heuristic based approach. These two features allow swam intelligence to search solutions both globally and locally. Global searching and local searching in solution space is called as exploration and exploitation, respectively. Till date, swarm intelligence (SI) like particle swarm optimization (PSO) [6], artificial bee colony (ABC) [7], ant colony optimization (ACO) [8], genetic algorithm (GA) [9] have been proposed to solve RNPP, but without considering network partition problem (NPP) in the network. We propose a novel solution, which minimizes the count of RNs in the network, by using a Grey wolf optimizer algorithm (GWO) [10]. As per our view, till date, no one has attempted to solve RNPP for segmented networks by using GWO. The remainder of the paper is organized as follows: Sect. 2 presents related works. Section 3 shows our proposed problem. Section 4 discusses the basis concept of GWO. Implementation of the proposed solution and the results of the simulations are shown in Sects. 5 and 6, respectively. Finally, Sect. 7 concludes the paper and discusses the future work.

2 Literature Review

In this section, we have shown a survey on previously attempted solutions for RNPP. RNPP techniques can be classified into four categories, i.e. (algorithmic-based, approximation based, heuristic based, swarm intelligence based) [2]. In this paper, we discuss only two of them (because of space constraint).

2.1 Swarm Intelligence Based Solutions

The authors of [6] have been implemented PSO to tackle the problem of RNPP in static hybrid networks. Particle swarm optimization is adopted with the integer planning problem to reduce the solution space. The results show that an optimal deployment of RNs largely improve energy efficiency and location of BS has a direct impact on results. Hashim [7] proposed an approach to increase the energy efficiency of WSN by deploying RNs using ABC. The key idea behind the usage of ABC is to optimize network parameter and extend the network lifetime with the minimum number of RNs. Kamal [8] jointly solve the problem of RNP and trajectory calculation for a mobile data collector (MDC) in hierarchical WSN using ACO SI. The proposed ACO based approach considers the model of traveling salesman problem (TSP). In [9], Azharuddin and Jana suggest an evolutionary GA-based technique of RNPP in hierarchical WSN. Initially, they consider a set of potential locations and seek to minimize RNs using GA based operations (crossover and mutation). Each potential location is considered as genes, which can hold two values 0 and 1. If a particular location is selected, it is marked as 1 otherwise 0. Performing efficient mutation and crossover with linear programming (LP) results in effective performance and generate optimized solutions.

2.2 Heuristic Solutions

Senel et al. [11] suggest an approach which places RNs towards the estimated center of mass of segmented area of the network. They did not consider the concept of minimum Steiner tree (MST), due to the creation of cut-vertex, which results in the partition of network again and again. Spider-web heuristic based approach [10] is considered to solve RNPP which guarantee inward placement of relays with better coverage and connectivity. In [12], Chen et al. proposed an approach for the application of WSN in wind farm monitoring. Their proposed approach strives to place RNs to connect sub-networks and get back inter-sub-network lost connectivity. The authors of [13] considered the working area as a grid with equal sized cells. Each segmented part of the network is represented by representative nodes. In [14] Bhattacharya et al. tackle the problem of placing RNs as well as BS, to get predefined performance objectives, which is called as multi-sink Steiner network minimum cost-Hop constraint (MSS-MCHC). The idea behind the placement of multiple sinks is to handle the network design scalability. Their proposed work includes a polynomial time approximation based algorithm (SmartSelect) which aims to minimize the cost of placing BSs and RNs. Based on the results of SmartSelect another heuristic algorithm is selected to improve the previous solution by destroying current one and explore solution space iteratively.

3 Problem Description

A flat structure of WSN is considered with random deployment of SNs. Initially, each SN is able to communicate with other SN using a wireless medium. The base station or sink node is established to observe the smooth working of the network. SNs transmit the sensed information using any predefined routing policy. When a number of SNs stops working due to their low battery power or damage their hardware due to an explosion on the battlefield, or any other issue, then segments are formed in the network. All these segments may not be able to transmit the useful information towards BS. Each segment holds a subset of SNs and the route to sink node may be disconnected. To tackle with this type of problem we suggest a meta-heuristic based approach to place RNs in the disconnected part of the network. Our problem can be defined as: *"WSN consists of a large number of tiny SNs. When any cut-vertex sensor node or any segmented group(s) of SNs is/are not reachable from sink node or from any other group, then the partition is formed. Network partitioning can be happened on a large scale, where a large number of SNs gets failed. Suppose N_{Seg} number of segments are created and each segment is represented by a single RN. Therefore, we have N_{Seg} number of points in the 2-D plane where the communication range of RN is considered as R. Our proposed solution address the problem of placing RNs in the damaged area to restore lost connectivity with a minimum count of RNs."*

4 The Grey Wolf Optimizer

Grey wolf optimizer (GWO) algorithm is a discrete nature swarm intelligence technique. It is inspired from the social and attacking behavior of grey wolves. Grey wolves are a special member of Candie family and prefer to live in a flock of size 5–12. The population of grey wolves is divided based upon a hierarchy. The one on the top of hierarchy dominates lower ones and so on.

The social hierarchy, shown in Fig. 1, is based on the dominance of wolves [10]. At the top of the hierarchy, a leader is present. The leader may be female or male, but the leader must have the ability to guide the group members. Wolves, at the top position of hierarchy, are called as alpha wolves (α). Alpha wolves have the privilege to make decisions and convey orders to all other group members. The second position in wolf social hierarchy is reserved for Beta wolves (β). Beta wolves take orders from alphas and convey the same to lower level members. Betas help alphas to make the decision about group management activities. In the case of unavailability of alphas, beta leads the other group members. At the third position, Delta wolves (δ) are present. Deltas follow the order of betas and alphas and give the order to omegas. Deltas have been assigned the role of, scout, hunter and they are the most experienced members of the family. They help another member during hunting and arrange food. Omega wolves (ω) available at the bottom of the hierarchy. Omegas follow the order given by the other wolves and acknowledge the same. Omegas are dominated by all others in the group of wolves. They get the food at last when all others have been finished. Grey wolves attack prey in a well-planned manner and forward gradually towards the location of prey. The same hunting procedure can be used to solve optimization problems. We also follow the same procedure to solve RNPP in an efficient manner. Grey wolves complete the hunting process in steps, which is shown below, suggested by [15].

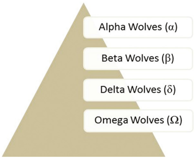


Fig. 1. Social hierarchy of gray wolves

- Trailing and tracking
- Surrounding the prey
- Aggress the prey

The mathematical modeling of hunting procedure [10] is shown in the next part of the paper. Our proposed algorithm uses the same concept to find an optimum location of RNs for a given set of representative nodes.

4.1 Surrounding the Prey

When the group of grey wolves encircles the prey, following mathematical model is used.

$$\vec{\mathrm{D}} = \left|\vec{\mathrm{R}}.\overrightarrow{\mathrm{L_P}}(\mathrm{i}) - \vec{\mathrm{L}}(\mathrm{i})\right| \tag{1}$$

$$\overrightarrow{\mathrm{L}}(i+1) = \overrightarrow{\mathrm{L_P}}(i) - \vec{\mathrm{Q}}.\vec{\mathrm{D}} \tag{2}$$

In Eq. 1 and 2, $\vec{\mathrm{D}}$ represent distance, $\vec{\mathrm{L}}\mathrm{p}$ and $\vec{\mathrm{L}}$ denote the location vector of prey and wolf respectively, $\vec{\mathrm{R}}$ and $\vec{\mathrm{Q}}$ are coefficient vectors which helps to make a good surrounding for prey and *i* represents the iteration number. Next two equations are used to generate both coefficient vectors $\vec{\mathrm{R}}$ and $\vec{\mathrm{Q}}$.

$$\vec{\mathrm{R}} = 2.\vec{\mathrm{r}}1 \tag{3}$$

$$\overrightarrow{\mathrm{Q}} = 2.\overrightarrow{\mathrm{q}}\,\overrightarrow{\mathrm{r}} - .\overrightarrow{\mathrm{q}} \tag{4}$$

In Eqs. 3 and 4, $\vec{\mathrm{r}}1$ and $\vec{\mathrm{r}}2$ are random vectors in the range [0, 1] and $\vec{\mathrm{q}}$ is decreased from value 2 to 0.

4.2 Aggress the Prey

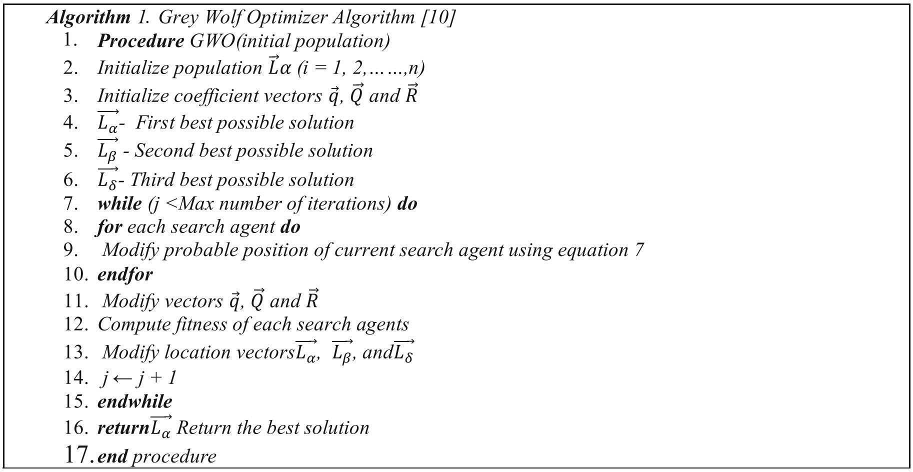
Attacking or aggressing the prey requires establishing a specific model to apply GWO to solve optimization problems. In this section, we give a generalized modeling for alpha, beta, delta and omega based on the study of authors [10]. Mathematical modeling for alpha, beta, and delta are given as:

$$\overrightarrow{\mathrm{D}_\alpha} = \left|\overrightarrow{\mathrm{R}_1}.\overrightarrow{\mathrm{L}_\alpha} - \vec{\mathrm{L}}\right|, \overrightarrow{\mathrm{D}_\beta} = \left|\overrightarrow{\mathrm{R}_2}.\overrightarrow{\mathrm{L}_\beta} - \vec{\mathrm{L}}\right|, \overrightarrow{\mathrm{D}_\delta} = \left|\overrightarrow{\mathrm{R}_3}.\overrightarrow{\mathrm{L}_\delta} - \vec{\mathrm{L}}\right| \tag{5}$$

$$\overrightarrow{\mathrm{L}_1} = \overrightarrow{\mathrm{L}_\alpha} - \overrightarrow{\mathrm{Q}_1}.\overrightarrow{\mathrm{D}_\alpha}, \overrightarrow{\mathrm{L}_2} = \overrightarrow{\mathrm{L}_\beta} - \overrightarrow{\mathrm{Q}_2}.\overrightarrow{\mathrm{D}_\beta}, \overrightarrow{\mathrm{L}_3} = \overrightarrow{\mathrm{L}_\delta} - \overrightarrow{\mathrm{Q}_3}.\overrightarrow{\mathrm{D}_\delta}, \tag{6}$$

$$\vec{\mathrm{L}}(\mathrm{i}+1) = (\overrightarrow{\mathrm{L}_1} + \overrightarrow{\mathrm{L}_2} + \overrightarrow{\mathrm{L}_3})/3 \tag{7}$$

The pseudo-code of GWO is shown below in Algorithm 1.



Algorithm *1. Grey Wolf Optimizer Algorithm [10]*

```
Procedure GWO(initial population)
Initialize population $\vec{L}\alpha$ (i = 1, 2,……,n)
Initialize coefficient vectors $\vec{q}$, $\vec{Q}$ and $\vec{R}$
$\overrightarrow{L_\alpha}$- First best possible solution
$\overrightarrow{L_\beta}$ - Second best possible solution
$\overrightarrow{L_\delta}$- Third best possible solution
while (j <Max number of iterations) do
for each search agent do
  Modify probable position of current search agent using equation 7
endfor
  Modify vectors $\vec{q}$, $\vec{Q}$ and $\vec{R}$
Compute fitness of each search agents
  Modify location vectors$\overrightarrow{L_\alpha}$, $\overrightarrow{L_\beta}$, and$\overrightarrow{L_\delta}$
$j \leftarrow j + 1$
endwhile
return$\overrightarrow{L_\alpha}$ Return the best solution
end procedure
```

5 Our Proposed Solution

In this section, we have discussed our proposed solution based on the GWO. We have mapped our problem using mathematical model of GWO. Initially, the set of segments S act as a flock of Grey Wolves and used as input for GWO Algorithm 1. The output of GWO can be somewhere in the damaged region such as all wolves (segments) encircle the prey (potential RN) position. Our proposed idea strives to populate RNs inwards center of the damaged area and strives to restore inter-segment lost connectivity. Our proposed solution is divided into different phases. Each phase gradually populates RNs and restore the segmented disjoint network. All of the phases are listed below:

i. Find the relative position of representative node
ii. Neighbor discovery
iii. Placed the RNs using GWO
iv. Termination phase

For better understanding, we discussed our proposed solution with an example to demonstrate the working of different phases. Step-by-step and detailed explanation of the proposed solution is shown in the next subsections.

5.1 Find the Relative Position of Representative Node

The process of repairing lost connectivity is started with failures of a large scale of SNs. During the normal working of WSN, when BS recognizes a sudden decrease in the number of packets received, then it is considered by the BS that segmentation/partition has been occurred in the network. At this point of failure, there is a need to execute a RNPP algorithm to repair the lost connectivity of the network. In the first phase of our proposed solution, we find a relative position of the representative node. Every segment is denoted Segi where i is $0 \leq i \leq N_{Seg}$ and N_{Seg} is total number of segments in the network. For experimentation purpose, the whole segment is denoted as a single X-Y coordinate location. Figure 2 shows the seven disconnected segments where 2-D coordinate of all segments are shown as: Seg1: (39, 77), Seg2: (68, 58), Seg3: (47, 15), Seg4: (2, 65), Seg5: (41, 54), Seg6: (7, 71) and Seg7: (69, 5). The above locations are calculated in the real simulation. The X-Y location is used as representative node for respective segments. Throughout this work, we use Euclidean distance to find the smallest distance between any two points. The formula used to find the Euclidean distance is stated below in Eq. 8.

$$|p_1, p_2| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{8}$$

where p_1 and p_2 represent two points, $|p_1, p_2|$ represents the Euclidean distance, and (x_1, y_1) and (x_2, y_2) are the X-Y coordinate of p_1 and p_2, respectively.

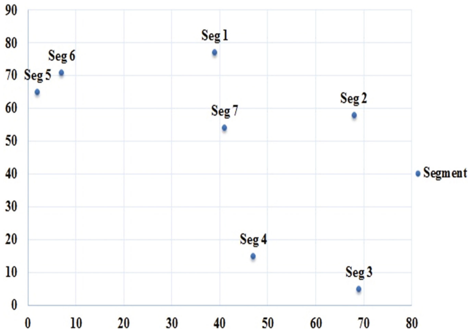


Fig. 2. Representative node of each segment is denoted by Segi

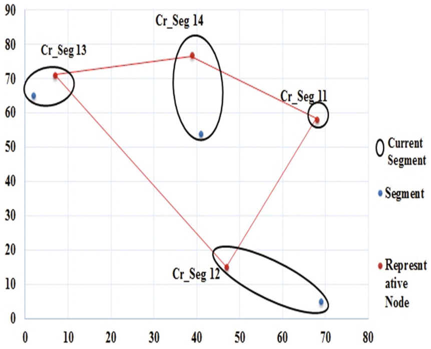


Fig. 3. Every segment join with its neighbor where Rr = 15 m

5.2 Neighbor Discovery

The second phase is suggested to find segments which are in the proximity of each other. Therefore, the proposed algorithm initiates the neighbor discovery process. This phase is executed for every disconnected segment. We assume that the representative node of every segment is an RN for the sake of simplicity. Due to this, it helps us to reduce the total number of calculations. To find neighboring segment Seg_n of segment Seg_i, it must satisfy equality $|Seg_i, Seg_n| \leq 2 * R$. The neighboring segments are shown in Fig. 4, where Seg_2 has no neighboring segment, therefore it is converted into segment Cr_Seg11, Seg_4 combines with Seg_3 and create segment Cr_Seg12, Seg_6 combines with Seg_5 and create segment Cr_Seg13, and Seg_1 combines with Seg_7 and create segment Cr_Seg14. Figure 3 also shows a convex hull generated over current representative nodes. The current round segments are denoted by Cr_Segij where i shows the iteration number and j denotes the segment number.

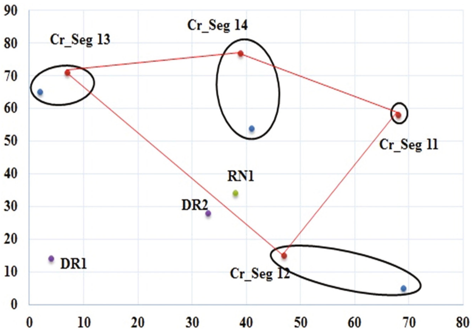


Fig. 4. First RN RN1 (38, 34) is placed

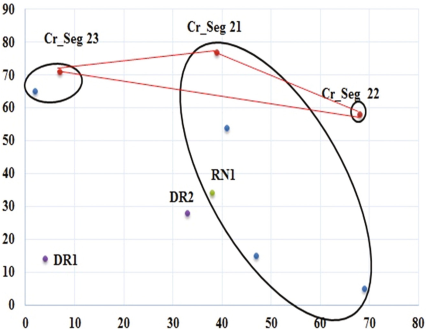


Fig. 5. Relay node RN1 connect two segments into single one

5.3 Placed the RNs Using GWO

This phase gradually restores lost connectivity and connect disjoint segments with each other. The third phase completes in several iterations until two or less than two segments are left. On each iteration, one optimized location of RN is calculated by GWO and placed RN at that location helps to restore connectivity. Optimized location given by GWO always be inside the convex hull. As current segments change, it leads to the creation of a new convex hull. The convex hull is generated using a Graham Scan algorithm proposed in [16]. Considering, one node as a representative for all other nodes in a segment. Further, a convex hull is shown over only these representative nodes. Figure 4 shows 3 new nodes, one is RN1 (38, 34) i.e. optimized location of the first relay node and other two are DR1 (4, 14) and DR2 (33, 28) denotes the discarded location because of, these are outside of convex hull. It is a side effect of meta-heuristic based solution of their random nature. This side effect can be removed with the help of the convex hull algorithm. Figure 5 shows how a relay node RN1 helps to restore lost inter-segment connectivity. Here, segment Cr_Seg12 connect with segment Cr_Seg14 and create segment Cr_Seg21.

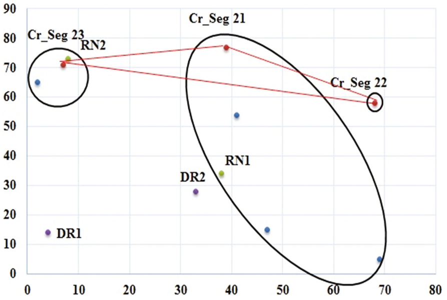


Fig. 6. Second RN RN2 (8, 73) is placed

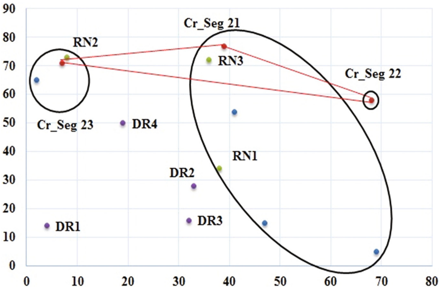


Fig. 7. Third relay node RN3 (36, 72) is placed

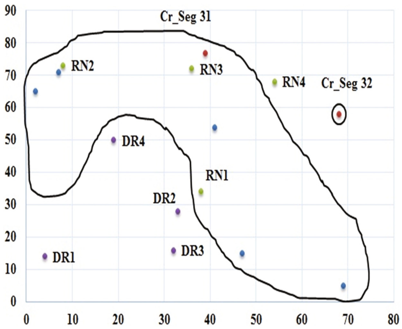


Fig. 8. Only two segments are left and Relay node RN4 (54, 68) is placed

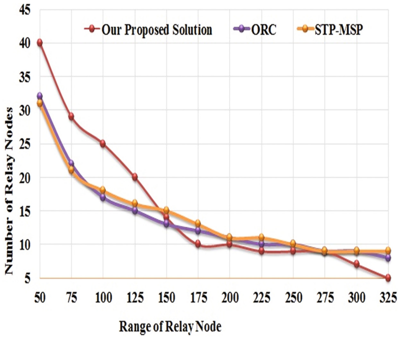


Fig. 9. Comparison for varying range with number of segment is $N_{Seg} = 5$

Figure 6 is the outcome of the second iteration where one more relay node RN2 (8, 73) is placed. But this does not create a connection between any two segments. Relay node RN2 lies on the line of the respective convex hull. In the third iteration of the proposed algorithm, the location of third relay node RN3 (36, 72) is found using GWO. Figure 7 shows two more discarded location DR3 (32, 16) and DR4 (19, 50). Sometimes, GWO generates locations which are not so relevant. Therefore, we discarded this type of locations using convex hull.

5.4 Termination Phase

The last phase is the termination phase. Our proposed solution is terminated when two or less than two segments are left in the WSN. The termination condition is inspired by the feature of GWO, because it requires at least three sized population to optimize the result. Same is taken in our proposed solution. Figure 8 shows disjoint segments left after the placement of required relays. Further, last two segments are connected manually. The location of relays might be somewhere on the joining line of last two representative nodes. The required number of RN to connect last two segments is calculated with the help of equality $\frac{|p_1,p_2|}{2*R}$ where p_1 and p_2 are X-Y coordinates of last two segments. If only one segment is left, lost connectivity is restored successfully. The pseudo-code of our proposed solution is shown in Algorithm 2.

***Algorithm** 2. Our Proposed Solution*

```
procedure (Area, $N_{Seg}$, R)
Initialize 2-D array S and $N_{Relay} \leftarrow 0$;
for Every segment pair i and j
 if $| S[i] , S[j] | \leq 2 \times R_r$
      Connect $i^{th}$ and $j^{th}$ segment
 end if  end for
      while S have more than 2 points do
            Initialize $C_{Relay}$ [2] ;
 CH ← Compute a convex hull over S
 $C_{Relay} \leftarrow$ GWO (S) ;
 If CRelay lie inside convex hull CH
 $N_{Relay}$++ ;
 Initialize integer C Temp_Seg←0 ;
 Initialize and 2-D array Temp_Seg ;
 for i ← 0 to $N_{Seg}$
 if $| S[i] , C_{Relay} | \leq 2 \times R_r$
 Temp_Seg ← S[i];
 C_Temp_Seg + + ;
 end if  end for
 if C_Temp_Seg == 0
 Create a new Segment with current relay.
 else if C_Temp_Seg == 1
 Add current relay to respective segment
 Else
 for all elements in Temp_Seg
 Join together
 S =  S – Temp_Seg;
 end for end if
 end if end if
 end while
 if S have two point then
 Temp ←$| S[0], S[1] | / (2 \times R_r)$
 $N_{Relay} \leftarrow N_{Relay}$ + Temp
 end if
 return $N_{Relay}$
 end procedure
```

6 Comparison and Results Discussion

Here, we compare our proposed solution with other state-of-art approaches (ORC [13], STP-MSP [17]). The comparison and simulation results are an evidence of our work done. We have implemented our algorithm in JAVA environment. Our results show that proposed solution is outperformed others when the communication range is large. But due to the randomness of GWO, sometimes the performance of our proposed solution is degraded when transmission range is small. The comparison is done, on the basis of total number of RN counts required to restore lost connectivity.

Initially, we choose some simulation parameters which are listed in Table 1. The following parameters are used to evaluate the proposed solution: (a) Number of disjoint segments (N_{Seg}), (b) transmission range of relay node (R) and (c) total number of relay node (N_{Relay}). During simulations, we have taken 1000 m × 1000 m area having a varying number of disconnected networks. Number of segments may vary from 5 to 10

Table 1. Simulation parameters

Parameter	Value
Area	1000 m × 1000 m
Sensor nodes	100–500
Total number of partition	5–10
Communication range of RNs	50 m–325 m
Number of placed RNs	5–45

and count of RNs also varies with the number of segments. All simulation results have been taken into consideration after the average of 30 experiments. Therefore, it is a strong evidence to increase the confidence level about the proposed solution. ORC [13] and STP-MSP [17] is the baseline RNP approaches, choose for comparison.

6.1 Total Number of Relay Nodes

This metric is used to compare the effectiveness of the proposed approaches for comparison. Our main objective is to minimize the total number of relay nodes. In Fig. 9, we compare the results for varying communication range, but the number of segments is taken fixed i.e. 5. The transmission range of RN has a direct impact on the total count of RNs. The same scenario is depicted in Fig. 10 which shows the result of ORC, STP-MSP. This simulation shows the side effect of stochastic nature of GWO where our algorithm outputs more numbers of relays compared with others. In this, it has been observed that our proposed solution outperforms over ORC and STP-MSP in all cases if range of an RN is 200 m. The relationship between the number of segments and total RNs can be seen here also. The last simulation which we have conducted, is only for our proposed solution. We combine all three parameters (number of segments, communication range, the number of RNs) to show a scenario about the impact of the count of the segments on the final count of RNs. Figure 11 is a 3-D dimension of simulation where a clear view of the above said can be seen.

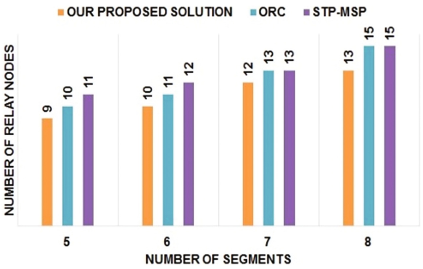


Fig. 10. Comparison for varying segments, where range of a relay node is R = 200 m

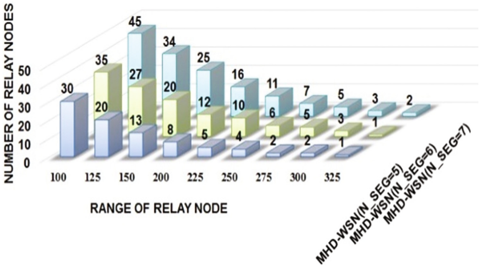


Fig. 11. Results of our proposed solution for varying number of segments

7 Conclusion and Future Work

WSNs have lots of applications in several scenarios, e.g. underwater, coal mine, battlefield, and forest) where the risk to human life is more. Therefore, the demand of WSNs increases for several real life applications. Further, the harsh surroundings become a problem for WSN, when the network is partitioned into several disjoint segments. At that time, there is a need to populate RNs which help to restore connectivity. To address this problem, we suggested a swarm intelligence based solution while considering GWO as a relevant SI to find the best positions of RNs. To reduce the impact stochastic nature of GWO, we embedded convex hull approach along with GWO while populating RNs inwards center of the damaged region. At last, we evaluate the performance gain of our proposed solution with some state-of-art solutions where we found good results. In the future, we check our proposed solution on the real test bed.

References

1. Ranga, V., Dave, M., Verma, A.K.: Network partitioning recovery mechanisms in WSANs: a survey. Wirel. Pers. Commun. **72**, 857–917 (2013)
2. Verma, A., Ranga, V., Angra, S.: Relay node placement techniques in wireless sensor networks. In: 2015 IEEE International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 1384–1389 (2015)
3. Ranga, V., Dave, M., Verma, A.K.: Node stability aware energy efficient single node failure recovery approach for WSANs. Malays. J. Comput. Sci. **29**(2), 106–123 (2016)
4. Ranga, V., Dave, M., Verma, A.K.: A hybrid timer based single node failure recovery approach for WSANs. Wirel. Pers. Commun. **77**(3), 2155–2182 (2014)
5. Ranga, V., Dave, M., Verma, A.K.: Relay node placement to heal partitioned wireless sensor networks. Comput. Electr. Eng. **48**, 371–388 (2015)
6. Zhao, C., Chen, P.G.: Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks. In: IEEE Congress on Evolutionary Computation. CEC 2007, pp. 3316–3320. IEEE, (2007)
7. Hashim, H.A., Ayinde, B.O., Abido, M.A.: Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm. J. Netw. Comp. Appl. **64**, 239–248 (2016)
8. Kamal, K.R.: Ant colony optimization for jointly solving relay node placement and trajectory calculation in hierarchical wireless sensor networks (2014)
9. Azharuddin, M., Jana, P.K.: A ga-based approach for fault tolerant relay node placement in wireless sensor networks. In: 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), pp. 1–6. IEEE, (2015)
10. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
11. Senel, F., Younis, M., Akkaya, K.: A robust relay node placement heuristic for structurally damaged wireless sensor networks. In: IEEE 34th Conference on Local Computer Networks. LCN 2009, pp. 633–640. IEEE, (2009)
12. Chen, Q., Hu, Y., Chen, Z., Grout, V., Zhang, D., Wang, H., Xing, H., Improved relay node placement algorithm for wireless sensor networks application in wind farm. In: 2013 IEEE International Conference on Smart Energy Grid Engineering (SEGE), pp. 1–6. IEEE, (2013)

13. Lee, S., Younis, M.: Optimized relay node placement for connecting disjoint wireless sensor networks. Comput. Netw. **56**(12), 2788–2804 (2012)
14. Bhattacharya, A., Rao, A., Naveen, K., Nishanth, P., Anand, S., Kumar, A.: Qos constrained optimal sink and relay placement in planned wireless sensor networks. In: 2014 International Conference on Signal Processing and Communications (SPCOM), pp. 1–5. IEEE, (2014)
15. Muro, C., Escobedo, R., Spector, L., Coppinger, R.: Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. Behav. Proc. **88**(3), 192–197 (2011)
16. Graham, R.L.: An efficient algorith for determining the convex hull of a finite planar set. Inf. Process. Lett **1**(4), 132–133 (1972)
17. Cheng, X., Du, D.-Z., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. Wirel. Netw. **14**(3), 347–355 (2008)