International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

An Approach to Efficient Data Redundancy Reduction While Preserving the Full Coverage in a WSN

Razan S.M. Saadaldeen[1], Yousif E.E. Ahmed[2], Abdalla A. Osman[3]

[1]PhD Student, Wad Medani College of Medical Sciences & Technology, Wad Madani, Sudan
[2]Assistant Professor, Dept. of Computer Engineering University of Gezira, Wad Madani, Sudan
[3]Professor, Dept. of Computer Sciences University of Elahlia, Wad Madani, Sudan country

---

Abstract - *In Wireless Sensor Network (WSN) to ensure reliability, large numbers of sensor nodes deployed randomly to cover the same target. Each sensor node monitor more than one target, if one sensor is dead another sensor can cover that target; but at the same time, there is a wastes of energy because all sensors processing the same data which produce redundant data. Redundant occur when the number of sensors that monitor the same target is more than the required, so there is a need to eliminate the redundancy in the sensed data up to suitable level in order to maintain the energy conservation and reliability. This was done using genetic algorithm that reducing the number of working sensors to minimize the redundancy with coverage constrain to cover all targets. The tool used was the MATLAB, the simulation results shows that the algorithm work best in a different number of sensors and it reduce the number of working sensors without affect the full coverage. The algorithm also outperform EBNDNS and ECCA protocols in terms of reducing the number of working sensors.*

Key Words: Coverage, Genetic Algorithm, Redundancy, Targets.

1. INTRODUCTION

Redundancy means use more than needed sensor node in the area to ensure reliability in case if a sensor node is dead another sensors can monitor the region and the WSN becomes more and more fault tolerant.

Redundancy can be classified into spatial redundancy, information redundancy or temporal redundancy. The spatial redundancy means obtain information for a specific location from multiple sensors by waiting reports. The temporal redundancy used to enhance the accuracy of sensor nodes readings in order to increase reliability also called time redundancy which means perform the same action timely. The last type is the information redundancy used to reconstruct lost information by using the redundant data, e.g. extra bits [1] [2].

This paper develop a method for minimizing the number of working sensors and reducing redundant data with coverage constrain. The next section is devoted for how researcher used different technique to minimize redundancy.

2. LITERATURE REVIEW

A lot of researches proposed a methods for reducing redundancy with different techniques but in the same time preserving the coverage to some extent. Researcher in [3, 4] analysis and develop a mathematical model for completely and partial redundancy for large-scale sensors network they conclude that, when the number of neighbors for a sensor increased lead to increasing in the probability of complete redundancy in that sensor. ECCA protocol [5] was proposed based on multi-objective genetic algorithm in a density deployment nodes the algorithm compute the optimal sensor sets cover by select a minimum number of sensors to cover all network. The algorithm achieve balanced performance on different types of sensor models with maintaining high coverage rate.

Another researchers [6] provides EBNDNS protocol for a node sleep scheduling algorithm to solve energy- hole problem in the multi-hop communication by taking 2-hop neighbor information. This method uses distance between neighbor sensors instead of location information. They achieve balanced energy consumption under ensuring the coverage ratio. Deepa and Sharmini [7] detect the packet level redundancy to eliminate the packet redundancy using elimination hashing algorithm by Rabin Karp which focus on the identification and elimination of the packet level redundancy with less energy consumption thereby receiving the data with reduction of duplicity. A Fuzzy-based Redundancy Avoidance protocol (FBRA) was presented [8]. In FBRA the data transmission phase is modified which only one sensor node from a group of node will send the data to the base station, thus maintain long network lifetime. In [9] a multi-objective genetic algorithm was develop to minimize redundancy by using minimum number of sensor while preserving a full coverage. The algorithm achieved balanced performance on different types of detection sensor models and maintaining high coverage rate comparing to existing solutions. In [10] they produce redundancy detection protocol for area coverage control in heterogeneous wireless sensor networks the strategy based on a geometric method called crossing coverage to check the redundancy status of nodes with different ranges. They produced better results than some of the most relevant solutions. authors in [11, 12] proposed method for minimum sensor set that cover all target and maximize the lifetime, Chih in [11] used a genetic algorithm to improve lifetime of disjoint covers for all target can be covered but in their consideration the critical sensor

which means a sensor that cover number of target other sensor can't cover the algorithm can improve the performance of some existing heuristic algorithm by 16% and in [12] providing a linear mathematical model to composed a subsets of sensors that can completely cover all targeted for a non-disjoint set covers for lifespan management in WSN, optimal solutions were obtained considering the number of sensors.

Many researcher optimize redundancy by review the number of neighbors covers a sensor node without considering the target that the sensors covers, this paper focus on reducing redundancy using genetic algorithm for randomly deployed sensors that each sensor can monitor number of targets with optimal number of working sensors that reduce data redundancy in WSN while preserving the coverage required based on single objective method.

3. Problem Formulation

Regarding a WSN with a number of sensors n was deployed to cover a number of targets m, the problem is to find optimal number of working sensors that reduce data redundancy in WSN while preserving the coverage required based on single objective. As illustrated in Figure 1 a WSN with a number of sensor nodes n=5, randomly and redundantly deployed (to ensure reliability) to monitor a number of targets, m=7 in square field.

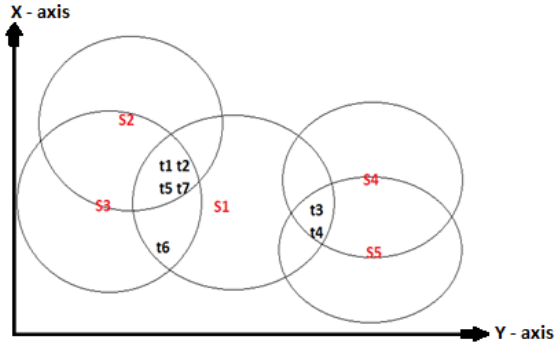

Fig -1: a WSN with sensors and targets.

Using the following assumption:-

- All Sensors are stationary after deployment.
- All sensors are homogeneous and have the same capabilities and initial energy (E_0).
- each sensor s_i could cover one or more target T_i), where i= 0…….n
- each s_i has target data redundancy:

$R_{si} = \sum_{i=0}^{n} s_i \cap s_n = T_i$, where { i= 0…….n} (1)

- the redundancy of the network for all sensors s_i is:

R= $\sum_{i=0}^{n} R_{si}$ (2)

The section below illustrates the method used to find the optimal solution to reduce data redundancy in WSN while preserving the coverage required.

4. THE METHODOLOGY

Based on the problem formulation any sensor must cover at least one target and the sensors that cover more than one target could be selected as the working sensors and other sensors could go to sleep. The methodology consist of two parts, the first part is developing a mathematical model for a wireless sensor network to calculate the data redundancy of each sensor in the network and the second part is using the genetic algorithm to minimize the redundancy by reduce the number of working sensors with the coverage constrain.

4.1 Mathematical Model

Firstly we need to formulate the individual cover relations matrix to express the relation between sensors and targets. For a random distributed set of sensors s and set of targets T to be monitored, each sensor s can cover a set of targets T(s) ⊆ T as follows:

$$S_{TS}=\begin{bmatrix} s11 & s12 \dots\dots s1n \\ s21 & s22 \dots\dots s2n \\ \dots & \dots\dots \dots \\ sm1 & sm2 \dots . smn \end{bmatrix} \quad (3)$$

$$s_{mn}=\begin{cases} 1 & if\ sensor\ j\ can\ cover\ target\ i \\ 0 & other\ wise \end{cases} \quad (4)$$

In the matrix S_{TS}, a row represents the targets T covered by sensor and each column represents the set of sensors S that can cover target.

For every target and sensor, there is randomly generated location coordinates, a sensors (x_s,y_s) and targets (x_t, y_t). Therefore, a target is covered by a sensor if the distance between them is less than or equal the coverage range r.

To obtain the distance (D) between the sensor and the target:

$$D=\sqrt{(x_t - x_s)^2 + (y_t - y_s)^2} \quad <=r \quad (5)$$

Equation (3) and Equation (4) can be used to calculate the coverage of all active sensors in the network, taking the union of the coverage:

Coverage (C)=$\cup_{i=0}^{n} (c_{si}).(AC)$ (6.1)

$$\cup_{i=0}^{n} c_{si} = T_1 \cup T_2 \cup T_3 \cup T_4 \dots\dots \cup T_n \quad (6.2)$$

$$AC=\begin{cases} 1 & if\ sensor\ i\ is\ active \\ 0 & other\ wise \end{cases} \quad (6.3)$$

Where c_{si} is the coverage of a sensor s_i.

To calculate the redundancy (R_{si}) for each sensor node (S_i) where a sensor have a redundancy if a targets that cover is monitored by another sensor:

$R_{si} = \sum_{i=0}^{n} s_i \cap s_n = T_i$ (7)

Where n is the number of sensor neighbor for that sensor.

From Equation (6) and Equation (7), to find the redundancy of the network for all sensors:

$R_n = \sum_{i=0}^{n} R_{si}$ *AC (8)

4.1.1 Calculating the redundancy and coverage

Based on the mathematical assumption and Figure 1 for 5 sensors deployed, let T = {t1, t2, t3, t4, t5, t6, t7} and S = {s_1, s_2, s_3, s_4, s_5}. Where targets cover by s_1 = {t1, t2, t3, t4, t5, t6, t7}, targets cover by s_2 = {t1, t2, t5, t7}, targets cover by s_3 = {t1, t2, t5, t6, t7}, targets cover by s_4 = {t3, t4} and targets cover by s_5 = {t3, t4}.

From Equation (3) cover relations matrix expresses relations between sensors and targets as follows:

$$S_{TS}=\begin{array}{c} \\ T1 \\ T2 \\ T3 \\ T4 \\ T5 \\ T6 \\ T7 \end{array}\begin{array}{c} s1 \quad s2 \quad s3 \quad s4 \quad s5 \\ \left[\begin{array}{ccccc} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array}\right] \end{array} \quad (9)$$

Form Equation (7) to acquire the redundancy for each sensor:

$R^{s_1}=\sum_{i=1}^{5} s1 \cap (s2, s3, s4, s5)$={t1,t2,t3,t4,t5,t6,t7 }=7 targets
$R^{s_2}=\sum_{i=1}^{5} s2 \cap (s1, s3, s4, s5)$ = {t1, t2, t5, t7} =4 targets
$R^{s_3}=\sum_{i=1}^{5} s3 \cap (s1, s2, s4, s5)$ = {t1, t2, t5, t6, t7} =5 targets
$R^{s_4}=\sum_{i=1}^{5} s4 \cap (s1, s2, s3, s5)$ = {t3, t4} =2 targets
$R^{s_5}=\sum_{i=1}^{5} s5 \cap (s1, s2, s3, s4)$ = {t3, t4} =2 targets

And from Equation (8) the redundancy of the network for all sensor:
So the summation of redundancy, $R^{s_i}=\sum_{i=1}^{5} Si$= R^{s_1}+ R^{s_2}+ R^{s_3}+ R^{s_4}+ R^{s_5}= 7+4+5+2+2=20

To get better performance the redundancy of the network will be divides by 2, and use it in the objective function of the algorithm, the reason for that if s1 and s2 is only active and the other sensors are sleep the redundancy will be 7+4=11 while s1 and s2 cover the same target, so the redundancy should be divide by 2 to be 11/2=5.5=5 to get near optimal redundancy. The actual redundancy s1$\cap$s2 in {t1, t2, t5, t7} =4 targets. To find the optimal solution by calculation that select the minimal number of working sensors to reduce the data redundancy for 5 sensors (s_1, s_2, s_3, s_4, s_5) each sensor can take value (0 or 1), 0: means the sensor is sleep and 1 means the sensor is active, so there are 2^5= 32 possible solutions, all possible solution will be estimated and investigated from (00001 to 11111). The best solution is (1 0 0 0 0) means s1 is active and the other sensors are sleep ($s_1(active), s_2(sleep), s_3(sleep), s_4(sleep), s_5(sleep)$) with coverage obtained = 7 and redundancy =0 .The calculation will runs 32 times to find the solutions for only 5 sensors. If there is 10 sensors, so there will be need for 2^{10} =100 possible solution so this is NP hard problem increase the number of sensor will need to increase the possible solution to find the exact one.

The problem presented is consumes exponential time: O (2^N) to find the exact solution which is NP hard problem. The genetic algorithm is needed to select the minimum number of sensors that have less data redundancy with full coverage. The Genetic algorithms are normally used in optimization and search problems for complex solution it used to reduce the search space. GA mostly requires solution representation of the genetic and a fitness function to evaluate the solution domain. The next section dedicated for the Genetic algorithm.

4.2 The Genetic algorithm

This section the proposed method laying on the GA as a useful method for finding the optimal solutions of complex problems. The basic steps for the GA: initialization, evaluation, selection, crossover, mutation and termination were adapted in the syntax of minimizing the redundancy. The number of working sensors must be minimized considering the full coverage as a constraint this is called single objective function where the objective function is to minimize the data redundancy and the fitness is the coverage.

Minimize R=$\sum_{i=0}^{ns}$ Rsi (10.a)
Subject to: $\cup_{i=0}^{ns} c_{si}$ >= T (10.b)
The genetic algorithm is needed to determine which sensors must be active and which are not active by the following steps.

4.2.1 Initialization

For GA initialization, create an initial population of gene, each gene in the chromosome can take the value (0 or 1), 0 for active and 1 for not active. The population is usually randomly generated and can be any desired size, from only a few individuals to thousands. The chromosome length is 1*n where n is the number of sensors (1*5).

4.2.2 Evaluation

Each member (row) of the population is then evaluated and calculates a 'fitness' for that individual. The fitness value is calculated by how well it fits with the desired requirements. The genetic algorithm is responsible for finding the minimal redundancy with constrain of the coverage. The coverage will be the fitness using Equations (10).

4.2.3 Selection

The roulette wheel selection is used as a genetic operator in genetic algorithm for selecting potentially useful solutions for recombination depending on the number of individuals in the population. The sensors with full coverage and minimal redundancy is selected.

4.2.4 Crossover

During crossover a new individuals will be created by combining aspects of selected individuals. The shuffle crossover and uniform crossover are used.

Shuffle crossover selects the two parents for crossover, randomly shuffles the genes in the both parents but in the same way, applies 1-point crossover technique by randomly selecting point as crossover point and the combine both parent to create two offspring. While, in uniform crossover each bit is chosen from either parent with equal probability.

4.2.5 Mutation

Mutation typically works by making very small changes at random to an individual's genome. The mutation is normally for one bit, and sometimes for number of bit. The mutation used in the research is for muted one bit by randomly changing a sensor in the chromosome from active to sleep and vice versa.

4.2.6 Termination

The GA termination could be by finding a good solution, by the number of generation or by timing. This research considers the number of generations. Different number of generations were used such as: {10, 20, 30 and 40}. Figure 2 Sums up the genetic algorithm process.

The GA is used to find optimal solution for find the minimum number of working sensors to reducing redundancy data with full coverage constrain. The GA helps in reduce the search space instead of use traditional solution with exponential time: O (2^N) which are classified as NP hard problem.

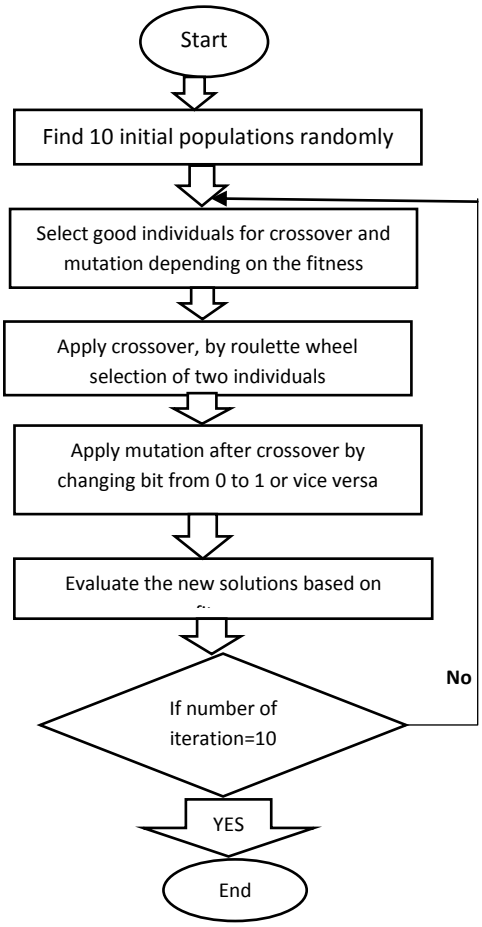

Fig -2: The process for genetic algorithm

5. RESULT AND DISCUSSION

This section presents the result of reducing redundancy algorithm by minimizing the number of working sensors. Several instances were used regarding the number of randomly deployed sensors, S| ∈ {5, 10, 15, 20}, the targets required to be monitored = 7.

For number of deployed sensor S= 5 to monitor 7 targets, let T = {t1, t2, t3, t4, t5, t6, t7} and S = {s1, s2, s3, s4, s5}.In the first scenario, after calculating the distance between sensors node and targets, the cover relation matrix T(s1) ={t1, t2, t3,t4, t5, t6, t7}, T(s2)={t1, t2, t5,t7}, T(s3)= {t1, t2, t5, t6, t7}, T(s4)= {t3, t4 }, T(s5)= {t3, t4 }.

S1 cover all targets, so if S1 active and other sensors sleep the coverage is 7 and the redundancy is 1. On another hand if {S2, s4} active the coverage = 6.

The Genetic algorithm is used to get near optimal solution using Equations (10).

In the algorithm, for the number of sensor |s|=5, number of generation (iteration) used=10, the initialization is 10 random possible solution with chromosome size =1*5.

A after using GA, the number of working sensor is decrease from 5 sensor to 1 sensor obtaining a full coverage, the redundancy of the network is reduced from 10 to zero and the energy dissipation is reduce from 1.000000000000000e-03 to 4.000000000000000e-04 lead to 6.0000e-04 saving of energy.

For 10 number of deployed sensor to monitor 7 targets, 10 sensors deployed randomly to monitor 7 targets; each sensor can cover more than one target. The chromosome size =1*10, with the same number of iteration and 10 random initial population; using MATLAB. The obtained result is After using GA, the number of working sensor is decrease from 10 sensor to 2 sensor obtaining a full coverage, the redundancy decreased from 21 to none and the energy dissipation is reduce from 0.002129545565155to 4.000000000000000e-04 lead to 0.0017 saving of energy.

For 15 number of sensor to monitor 7 targets, 15 sensors deployed randomly to monitor 7 targets, each sensor can monitor more than one target. using GA the number of working sensors decrease from 15 sensor to 2 sensors with equivalent full coverage and minimal redundancy from 28 to 1 and the energy dissipation is reduce from 0.003229545565155 4.000000000000000e-04 lead to 0.0028 saving of energy.

For 20 number of sensor to monitor 7 targets, 20 sensors deployed randomly to monitor 7 targets. the number of working sensor is decrease from 20 sensor to 7 sensor with full coverage and minimal redundancy from 35 to 7 and the energy dissipation is reduce from 0.004000068812280 to 0.00140000000000000 lead to 0.0017 saving of energy.

The conclusion of S| ∈ {5, 10, 15, 20}, the targets required to be monitored = 7 are shown in Table 1.

Table -1: The result of the optimization for using different

No. of sensors	Redundancy of the network	Best redundancy obtained
5	10	0
10	21	0
15	28	1
20	35	7
Best coverage obtained	No. of working sensors	Energy saved
7	1	6.0000e-04
7	2	0.0017
7	2	0.0028
7	7	0.0026

number of sensors

Figure 3 shows the redundancy reduce for each number of sensors {5, 10, 15, 20}, form the first look it could be seen that there is more than half redundancy reduced.

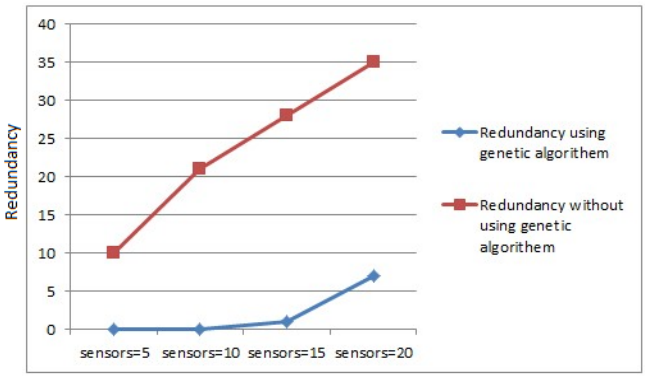

Fig -3: Redundancy according to number of sensors.

From Figure 3 when using 5 and 10 sensors the redundancy was reduced to 0%, for 15 sensors the redundancy reduced from 28% to 1% with using only 2 working sensors, also for 20 sensors the redundancy reduce from 35% to 7%. The redundancy is reduce to less than half with using less number of working sensors and preserving the full coverage.

Also in Figure 4 the number of working sensor decrease without affecting the coverage with S= {5, 10, 15, 20} and target =7.

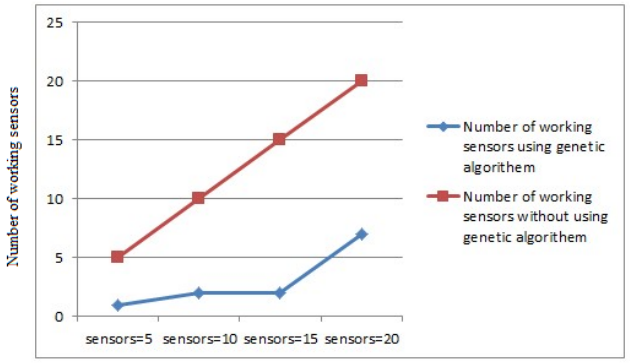

Fig -4: The number of working sensors with and without GA.

From Figure 4 it's clear that the number of working sensors reduced will conserve the full coverage. For instance the proposed algorithm discover that only required 1 sensors from 5 sensors to cover the all target and 2 sensors required in the case of 10 and 15 sensors deployed randomly.

Figure 5 depicts that the coverage has not been affected when reducing the number of working sensor.

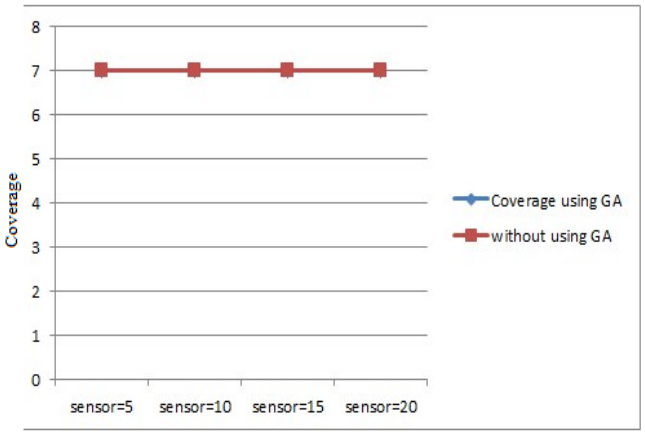

Fig -5: the number of working sensors with and without GA

From Figure 5 it could be notice that after reducing the number of working sensor and minimizing the redundancy more than the half all target all covered.

To conclude, a number of sensors, s= {5, 10, 15, 20} was used to cover 7 targets. The sensors deployed randomly, the proposed method using the optimization method genetic algorithm has been used with initial population size 10. The result prove that it could be reduce the number of working sensors and the redundancy to noticeable amount without affecting the network coverage.

The next section dedicated for shuffle and uniform crossover, since GA uses crossover and mutation to find the optimal solution, different type of crossover also used in the proposed method to find which one will be best to be use regarding with the problem.

5.1 GA Shuffle and uniform crossover

Shuffle crossover selects the two parents for crossover. It firstly randomly shuffles the genes in the both parents but in the same way. Then it applied 1-point crossover technique by randomly selecting point as crossover point and the combine both parent to create two offspring. In uniform crossover each bit is chosen from either parent with equal probability.

In the proposed GA two types of crossover are used separately with the same number of sensors=20 and targets=7. The overall redundancy of sensors and targets is 35, the number of generation is 10 and the initial population is 20 as shown in Figure 6 from the first view it's clear that the uniform crossover outer perform the shuffle crossover because it acquire minimum redundancy and less number of working sensors, although Shuffle crossover find a minimum redundancy at generation 1 but during generation uniform crossover achieve minimum redundancy than shuffle.

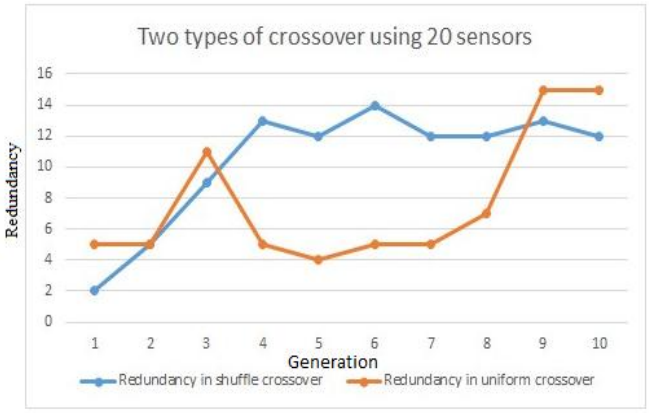

Fig -6: Uniform and shuffle from generation 1 to 10.

On the same hand, the shuffle also start with minimum number of working sensors but the number of worked sensors increase during generation. As shown in Figure 7.

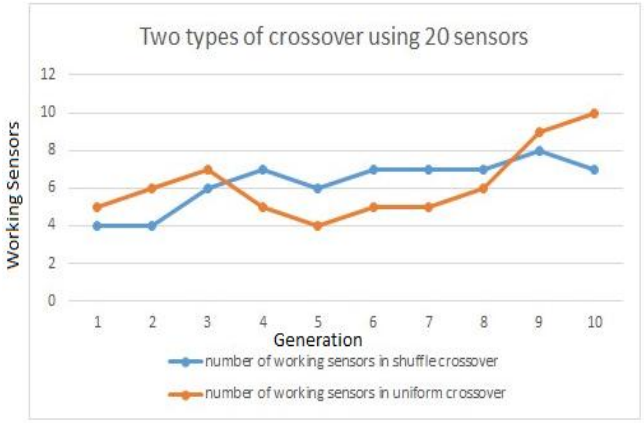

Fig -7: Comparing the number of working for uniform and shuffle crossover.

In Figure 7 uniform crossover use less number of working sensor than shuffle crossover. At the start, of the generation the shuffle crossover chose less working sensors but during generation using crossover ad mutation the number of sensors is increase.

In general, uniform crossover present noticeable result than shuffle crossover in the term of reducing redundancy and number of working sensor.

5.2 Using Different number of generation and initial population size

This section considers {20, 30 and 40} sensors deployed randomly in square region and GA with initial population size of {20, 30, 40 and 50} and the number of generation {10, 20, 30 and 40}. To envisage the best redundancy and number of working sensors will be obtained.

For 20 sensors deployed randomly with initial population 20 and different generation number {10, 20, 30 and 40}, in each run the generation number is changed to different value to obtain the best redundancy and the number of working sensors. The result obtained is shown in Table 2.

Table -2: Different generation with the same initial population

Number of sensors	Number of target	Generation number
20	7	10
20	7	20
20	7	30
20	7	40
Initial population	Best redundancy/35	Number of working sensor
20	12	10
20	7	7
20	9	6
20	9	4

As notice the number of working sensor is decrease when increase of the generation number, this mean increasing the number of generation will obtain better result. The best redundancy was introduced by using 20 generation and 20 initial population.

Table 3 illustrates the best redundancy for the same 20 sensors randomly deployed, with initial population equal {20, 30, 40 and 50} and the same generation number.

Table -3: Same generation with the different initial population

Number of sensors	Number of target	Generation number
20	7	20
20	7	20
20	7	20
20	7	20
Initial population	Best redundancy/35	Number of working sensor
20	7	7
30	8	4
40	8	3
50	2	2

As cleared, the number of working sensors also decrease while increasing the size of the initial population. But there is better result than Table 2. The best redundancy and number of working sensors appear when using initial population=50.

Figure 8 below depicts the redundancy of the network without using genetic algorithm which is the blue line, and the redundancy of the network when using genetic algorithm and change in the generation and initial population, different redundancy obtained.

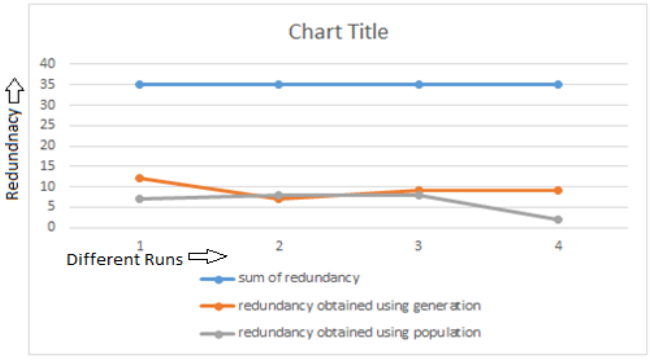

Fig -8: redundancy obtained using different generation and population.

From Figure 8 it's clear that the redundancy is less when using increasing the population size.

Figure 9 below depicts the number of working sensor used to cover all target without using genetic algorithm which is the blue line, and the number of working sensor used to cover all target when using genetic algorithm and change in the generation and initial population, different number of working sensors is obtained by the algorithm.

Fig -9: minimize of 20 sensors using different generation and population.

Figure 9 cleared that the number of working sensors was minimal when increasing in the population size.

As general, increasing the population size obtain better result in reduce redundancy and number of working sensors.

For 30 sensors deployed randomly with initial population 30 and different generation number {10, 20, 30 and 40}, each run the generation number is changed to different value to obtain the best redundancy and the number of working sensors. The result obtained is shown in Table 4

Table -4: 30 Sensors with the same initial population and different generation.

Number of sensors	Number of target	Generation number
30	7	10
30	7	20
30	7	30
30	7	40
Initial population	Best redundancy/49	Number of working sensor
30	11	7
30	12	8
30	9	6
30	14	10

From Table 4 it's clear when increasing the number of generation the algorithm didn't obtain better performance in the term of number of working sensors or redundancy, when the generation is 30 it's get better performance but the redundancy increased a gain in 40 generation.

For 30 sensors deployed randomly with different initial population {20, 30,40 and 50} and the same generation number, in each run the generation number is used with different value to obtain the best redundancy and the number of working sensors. The result obtained is shown in Table 5.

Table -5: 30 Sensors deployed randomly with the same generation and different population size.

Number of sensors	Number of target	Generation number
30	7	30
30	7	30
30	7	30
30	7	30
Initial population	Best redundancy/35	Number of working sensor
20	18	13
30	9	6
40	12	6
50	12	9

In each run its notice that, the redundancy obtained and the number of working sensors is reduced through changing the initial population. Reaching its lowest when the initial population is 30. This result outperform the result of Table 4.

Figure 10 depicts the number of working sensors used to cover all target without using genetic algorithm which is the blue line, and the number of working sensors used to cover all target when using genetic algorithm and change in the generation and initial population, different number of working sensors is obtained by the algorithm.

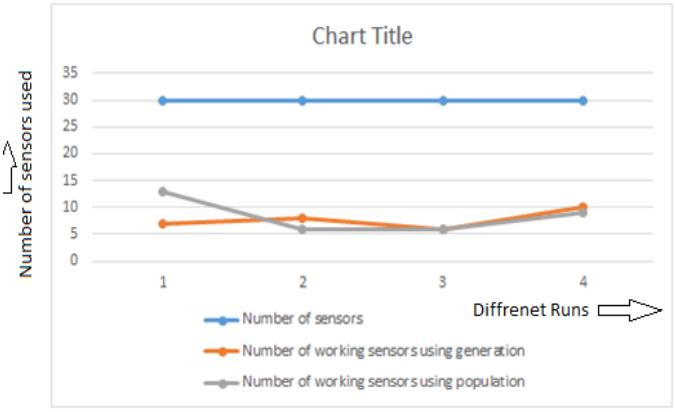

Fig -10: different generation and population size using 30 sensors

It's cleared that changing in the generation acquire better result in term of redundancy and number of working sensors.

From Figure 11 the redundancy obtained during changing in the generation and the population size in each run. The blue line is the redundancy of the network for 30 sensors when all sensor are active.

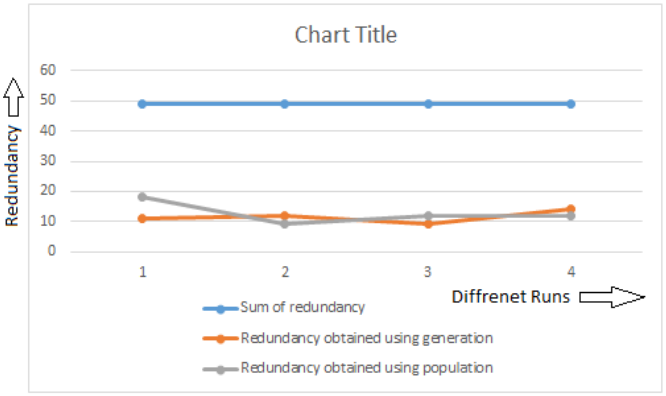

Fig -11: the redundancy of the network with different generation and population size.

As notice, there is no a measurable different, but as general changing in generation get better result than population.

For 40 sensors randomly deployed, in each run the generation number is different {10, 20, 30 and 40} with using the same initial population. As illustrated in Table 6.

Table -6: 40 sensors deployed randomly with different generation and the same initial population

Number of sensors	Number of target	Generation number
40	7	10
40	7	20
40	7	30
40	7	40

Initial population	Best redundancy/88	Number of working sensor
40	18	13
40	25	12
40	23	10
40	26	13

As cleared, there is no better performance in redundancy during the generation change, started from its best values (redundancy and number of working sensors) using 10 generation, on the same side the number of working sensors maintain its level and reach its best value when using 30 generation it 40 population.

For 40 sensors randomly deployed, with the same generation number and different initial population {20, 30, 40 and 50}. As illustrated in Table 7.

Table -7: 40 sensors deployed randomly with the same generation and the different initial population.

Number of sensors	Number of target	Generation number
40	7	40
40	7	40
40	7	40
40	7	40
Initial population	Best redundancy/35	Number of working sensor
20	28	14
30	32	17
40	35	12
50	25	9

From Table 7 it can be cleared that the redundancy is increase and decrease during the change in the size of the initial population reaching its best value in the 50 initial population, but during that was taking high redundancy. On other hand the number of working sensor decrease near to half. As general, increasing the initial population obtained a mountable result. Figure 12 illustrate using 40 sensors with changing in generation and population according to the number of working sensors.

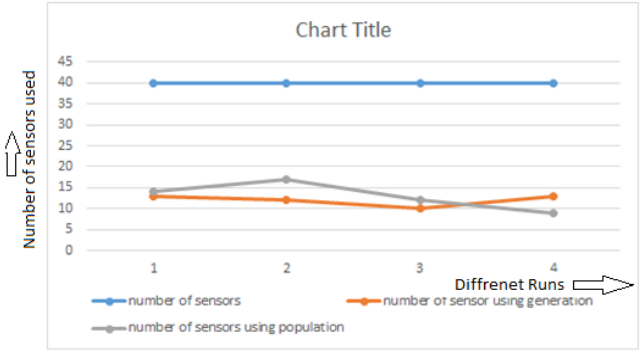

Fig -12: changing in generation and population size using 40 sensors

As general, using different generation number gives better result than using different population size. They both start with the same performance value but during runs, the generation outer perform the population.

Figure 13 the redundancy of the network for all sensors when 40 sensors.

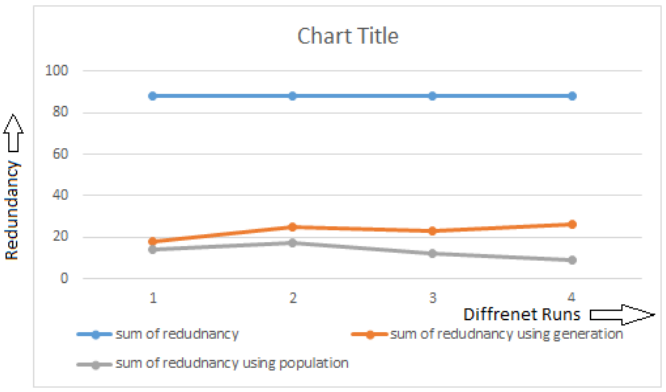

Fig -13: the redundancy of the network for 40 sensors.

As notice from Figure 13 at the beginning using different population its better comparing with using different generation.

To sum up, at the start when 20 sensors were deployed, using different population size was significantly better than using different generation, but in some cases the generation was as the same performance as the population size for example, when using 40 sensors it reduce the number of working sensors to a better result than different population size. All in all, changing in population size is the best than increasing the number of generation.

5.3 Comparing the algorithm with existing protocol using 100 and 200 number of sensors

There are many protocols have been used to reduce redundancy data in randomly deployed WSNs, such as Energy-efficient Coverage Control Algorithm (ECCA) and energy balanced non-uniform distribution node scheduling algorithm (EBNDNS). The proposed protocol was compared with such protocols using the same instances like number of deployed sensors, network size and the sensing radius.

The proposed protocol compared with ECCA protocol using 100 sensors node deployed randomly in a 100 * 100 square with sensing radius is set to 13. The ECCA perform full coverage using 70 sensors while the proposed method minimizing the number of working sensor to 37 with full coverage. Table 8 the comparing between ECCA and the proposed method.

Table -8: ECCA and the proposed method

No. of Generation	Number of Sensor deployed	No. of working sensors for The Proposed method
10	100	37
40	100	42
No. of working sensors for ECCA protocol	Coverage obtained in ECCA	Coverage obtained in proposed
70	10	10
55	9	10

As shown in Figure 14 the y-label is the number of deployed sensors. When using 40 generation, the number of sensors is also minimum than using the ECCA with maintaining the full coverage

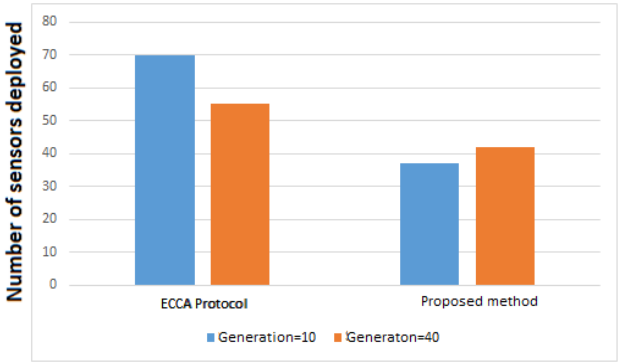

Fig -14: comparing the proposed protocol with EECA.

In Figure 15 another protocol is used for comparing the EBNDNS using two scenarios 100 and 200 number of randomly deployed sensors in 200m×200m square with sensing radius 15m. In 100 sensors scenario, the proposed system reduce the number of working sensors form 100 to 37 sensor achieving the full coverage while the EBNDNS use 60 sensors to cover the network. In 200 sensors scenario, the proposed system reduce the number of working sensors from 200 to 78 while EBNDNS used 105 sensor out of 200.

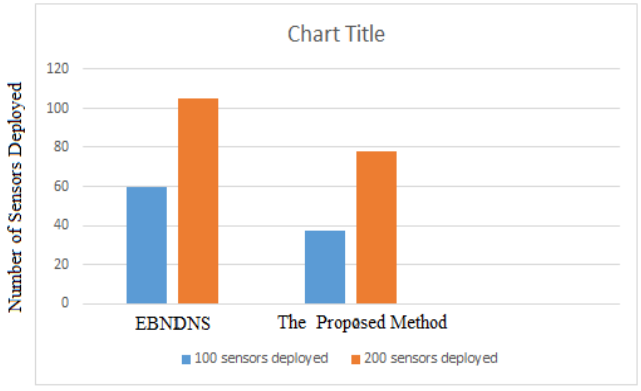

Fig -15: Comparing EBNDNS protocol with the proposed method.

From Figure 15 the proposed method achieve better results by reduce the number of working sensor near to half when using 100 sensors and to noticeable amount when using 200 sensors, while preserving the same full coverage.

To sum up, the proposed method outer perform the existing solution in term of reducing the number of working sensors and maintain the full coverage.

6. CONCLUSION

In conclusion, the algorithm reduce the number of working sensors up to adequate level to minimize the data redundancy while preserving the full coverage, this is done using Genetic Algorithm by selecting the sensors that cover large number of targets as the active sensors and other sensors went to sleeps, different number of sensors and targets were used to test the performance of the algorithm and the results are compared with existing solutions ECCA and EBNDNS using their same parameters. The proposed algorithm outer perform the existing solutions and can help in reducing the number of working sensors without affecting the full coverage.

7. REFERENCES

[1] I.F. Akyildiz, I.H. Kasimoglu, "Wireless sensor and actor networks: research challenges," Ad Hoc Networks Journal 2 (4) pp. 351–367, 2004.

[2] Curiac, D.I., Volosencu, C., Pescaru, D., Jurca, L. and Doboli, A., 2009. Redundancy and its applications in wireless sensor networks: a survey. WSEAS Transactions on Computers, 8(4), pp.705-714.

[3] Gao, Y., Wu, K. and Li, F., 2003, September. Analysis on the redundancy of wireless sensor networks. In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications (pp. 108-114).

[4] Wu, K., Gao, Y., Li, F. and Xiao, Y., 2005. Lightweight deployment-aware scheduling for wireless sensor networks. Mobile Networks and applications, 10(6), pp.837-852.

[5] Jia, J., Chen, J., Chang, G. and Tan, Z., 2009. Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. Computers & Mathematics with Applications, 57(11-12), pp.1756-1766.

[6] Shan-shan, M. and Jian-sheng, Q., 2014. Energy balanced non-uniform distribution node scheduling algorithm for wireless sensor networks. Applied Mathematics & Information Sciences, 8(4), p.1997.

[7] Priya, D. and Enoch, S., 2018. The Effect of Packet Redundancy Elimination Technique in Sensor Networks. JCS, 14(6), pp.740-746.

[8] Sharma, T., Kumar, B. and Tomar, G.S., 2013. FBRA: Fuzzy Based Redundancy Avoidance Protocol in Densely Deployed Wireless Sensor Network. International Journal of Smart Device and Appliance, 1(1), pp.1-16.

Jia, J., Chen, J., Chang, G. and Tan, Z., 2009. Energy efficient coverage control in wireless sensor networks based on multi-objective.

[9] Jia, J., Chen, J., Chang, G. and Tan, Z., 2009. Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. Computers & Mathematics with Applications, 57(11-12), pp.1756-1766.

[10] Zahmatkesh, A. and Yaghmaee, M.H., 2012. A genetic algorithm-based approach for energy-efficient clustering of wireless sensor networks. International Journal of Information and Electronics Engineering, 2(2), p.165.

[11] Lai, C.C., Ting, C.K. and Ko, R.S., 2007, September. An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In 2007 IEEE Congress on Evolutionary Computation (pp. 3531-3538). IEEE.

[12] Ahmed, Y.E.E., Adjallah, K.H., Kacem, I. and Babiker, S., 2015, October. Integer linear programming based scheduling method for wireless sensors network lifespan optimization. In 45th International Conference on Computers & Industrial Engineering 2015 (CIE45) (Vol. 2, pp. 1069-1076). Curran Associates, Inc.