The Research Monograph Series in Computing, Electrical & Communication Networks
Vol. 1 (01), May 2023, pp. 41–59

CHAPTER 3

HYBRID ROUTING SCHEME

3.1 INTRODUCTION

The purpose of WSN routing is to propose a least cost of route in fleet of reliable communication between sensor nodes to base station. Many researchers have undergone for solving the routing problem in swarm Wireless sensor network. Many routing algorithms are proved with P-compete. One of the most vital problems in wireless sensor networks is finding optimal routes for transmitting data between sources to destination. Generally it pairs in a multi-hop fashion. Several algorithms have been proposed for routing. When the stated problem is non-deterministic, if so it is under the problem of less ambiguity and deterministic and transfer rate be at polynomial time , then it is proved to NP-compete and run time execution (data transfer between sensor nodes to server) be at polynomial time. Particle Swarm Optimization technique is one of the peculiar computational intelligence which has the property of infinite size and easy flexible computational implementation and have solution for many optimal problem. PSO is inspired by the particle behaviour of external creatures (birds, bees, ants, fish etc). It is one of the stochastic based optimization model proposed by (Kennedy and Eberhart, 1995). A new family of algorithms emerged inspired by intelligence (SI). This provides a novel approach to distributed optimisation problems. The expression "swarm intelligence "defines any attempt to design algorithms inspired by the collective behaviour of social insect colonies and another animal societies. Swarm intelligence provides a basis with which it is possible to explore distributed optimization problems without explore centralised control or provision of global model. Initial research has unveiled a great deal of matching properties between the routing requirements of sensor networks and certain feature of SI. There are some notable algorithms which uses ant like mobile agents to maintain routing and topology discover for wireless sensor networks. In this work we order the nodes based on their energy efficiency and their focusing towards node path.

Swarm intelligence boasts a number of advantages due to the use of mobile agents they were denoted as follows:

1. Scalability: Population of the agents can be adapted consistently to the network size. Scalability is also promoted by local and scattered agent interactions.
2. Fault tolerance: Swarm intelligent processes do not rely on a centralized control mechanism. Therefore the loss of a few nodes or links does not result in catastrophic failure, but rather leads to graceful, scalable degradation.
3. Adaptation: Agents can alter, expire or replicate, according to network changes.
4. Speed: Changes in the network can be propagated very fast, in contrast with the Bellman-Ford algorithm.
5. Modularity: Agents act independently of other network layers.
6. Autonomy: Little or no human supervision is required.
7. Parallelism: Agent operations are inherently parallel. These properties make swarm intelligence very attractive for ad-hoc wireless networks. They also render swarm intelligence suitable for a variety of other applications, apart from routing, including robotics and optimization.

Evolutionary optimization schemes like genetic algorithms (GA) and PSO have successfully been used in the past to solve many NP-hard optimization problems. GA and PSO are similar in the way that both techniques are population based search schemes that mimic the natural biological evolution and/or the social behavior of species. Each member of the population represents a candidate solution to the problem addressed, and over time they evolve to represent some other candidate solution. One advantage of PSO over GA is that PSO is more computationally efficient some performance comparison studies between GA and PSO have been reported in the Figure. In a novel GA based scheme is proposed to solve dynamic RWA problem in wavelength routed optical networks. Genetic algorithms are swarm intelligence inspired search schemes based on the idea of natural selection and natural genetics. The member of the population (gene) represents a route from source to destination node i.e. a candidate solution to the routing sub-problem for DRWA. Genetic operators like crossover, mutation and then selection are applied to create a new population of genes. Ammar W. Mohemmed and Nirod Chandra Sahoo (2007) has proposed a novel hybrid algorithm based on PSO and a noising meta-heuristic for computing shortest paths in the network. The hybrid PSO based scheme shows better performance as compared to GA-based search algorithms for optimal shortest path computation. The GA algorithms are proposed for solving DRWA in all-optical WDM networks. In our work, the GA based schemes proposed for performance comparison purposes with our novel PSO-based algorithm.

3.2 HYBRID SCHEME 1 – OPTIMIZING LOCALIZATION ROUTE USING PARTICLE SWARM-A GENETIC APPROACH

3.2.1 Particle Swarm Optimization (PSO)

PSO is a population based optimization technique, developed by Kennedy and Eberhart (1995) inspired by social behavior of bird flocking (and schools of fish). In PSO, a swarm is a collection of particles where each particle has both a position and velocity. The position of the particle represents a candidate solution to the problem space while the velocity is used to move the particle from one position to another. The "classical" PSO equation where the position and velocity represents physical attributes of the particles is represented by (3.1) and (3.2)

$$V_{id} = V_{id} + n_1\, r_1\, (P_{id} - X_{id}) + n_2 r_2\, (P_{id} - X_{id})\ i=1,2...,N.d=1,2,..D \quad (3.1)$$

$$X_{id} = X_{id} + V_{id} \quad (3.2)$$

P_{id} is the personal best position, a particle has reached; $P_{id}n$ is the global best position of all the particles. η1 (the self-confidence factor) and η2 (the swarm-confidence factor) are positive constants called 'acceleration constants' to determine the influence of P_{id} and $P_{id}n$; r_1 and r_2 are independent random numbers in the range [0,1]. Is the total number of particles in the swarm and D is the dimension of the problem search space. PSO starts by randomly initializing the position and velocities of all the particles in the swarm over the problem space. The position of i^{th} particle is represented by the vector $X_i = [X_{i1} + X_{i2}...X_{id}]$ and velocity of ith particle is represented by the vector $V_i = [V_{i1} + V_{i2}...V_{id}]$, where D is the number of function parameters being optimized. For each iteration (until the convergence criteria is met), the fitness function is applied to the particles to quantize their respective positions over the problem search space. The particle among the finest fitness value in the neighborhood is marked as the global/local best particle. Each particle will also keep a record of its personal best position searched so far. Equation (3.1) is used to calculate new velocity for each particle in the swarm based on particle's preceding velocity, its current and personal finest position, and the position of the particle with best fitness value in the neighborhood. Equation (3.2) is then used to apply the velocity to the particle. As an outcome of this, the particle will move regarding to a new position i.e. it will now correspond to a new aspirant resolution to the problem being studied.

3.2.2 Genetical Swarm Optimization

Some comparisons of the performances of GA and PSO are present in the literature, underlining the reliability and convergence speed of both methods, but continuing in keeping them separate. Anyway, the population-based representation of the parameters that characterize a particular solution is the same for both the algorithms; therefore it is possible to implement a hybrid technique in order to utilize the qualities and uniqueness of the two algorithms. Some attempts have been done in this direction with good results, but with weak integration of the two strategies, because one algorithm is used mainly as the pre-optimizer for the initial population of the other one.

The hybrid technique here proposed, called Genetic Swarm Optimization (GSO), is essentially a population-based heuristic search technique which can be used to solve combinatorial optimization problems, modeled on the concept of natural selection but also based on cultural and social evolution. GSO algorithm consists in a strong cooperation of GA and PSO, since it maintains the integration of the two techniques for the entire run. In each iteration, the population is divided into two parts and they are evolved with the two techniques in that order. They are then remerged in the modernized population, that is yet again divided erratically into two parts in the next iteration for another run of genetic or particle swarm operators. The population revise concept can be effortlessly understood thinking that a part of those individuals is substituted by new generated ones by resources of GA, while the enduring are the same of the earlier generation but moved on the solution space by PSO.

3.2.3 Flooding in GA

The driving constraint of GSO algorithm is the Hybridization Coefficient (HC); it express the percentage of population that in each iteration is evolved with GA: so HC=0 means the procedure is a pure PSO (the whole population is updated according to PSO operators), HC=1 means pure GA, while 0<HC<1 means that the corresponding percentage of the population is developed by GA, the rest with PSO.

3.2.4 Pseudocode for the PSO-GA

Function GA=PSO(F, fit, i ,m,h)
Begin
Initialize particle do
For each particle
Calculate fitness function of the particle i(m)
If i(m)) is better than Ffit
set current value as the new Ffit
End_ For
Set hfitto the best fitness of ∀ particles
For ∀ particle
Calculate particle rate according $V_{id} = V_{id} + n_1 r_1(P_{id} - X_{id}) + n_2 r_2(P_{id} - X_{id})$
Update particle position according equation $X_{id} = X_{id} + V_{id}$
End_For
Check ∀ particle
For ∀ iteration
Generate Local criterian for hfit
Set Ffit for maximum
Calculate connection Matrix
Calculate Fitratio
End_While when maximum recursions attained
End

3.2.5 Implementation Results

The proposed hybrid scheme 1 was implemented using MATLAB. The results are shown in Figures 3.1–3.4. In this Figures, the estimated position of PSO algorithm is much closer to the actual source position, the estimation error between the actual position and the estimated position is about 1.3m. The estimated position will approach to the actual position with the increasing number of sensor nodes and signal-noise-ratio through the information of experiments, and while the number of sensor nodes is small the estimated position of PSO algorithm is more accurately than other searching algorithms.

3.3 HYBRID SCHEME 2 – SWARM-CLUSTER BASED ROUTING SCHEME FOR HETEROGENEOUS NODES

3.3.1 Node Identity

Identifying the node and its pattern is very important to save the node identity in terms of data routing, S-N communication

```
>> ConnectionMatrix

ConnectionMatrix =

  Columns 1 through 19

     0     0     0     0     0     1     0     0     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     0     1     1     0     0     1     0     0     1     0
     0     0     0     0     0     0     0     0     1     0     1     0     0     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     1     0     1     1     0     0     0     0     0     0     0
     1     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     1     0
     0     0     0     0     0     0     0     0     0     1     0     0     0     1     0     0     1     0     0
     0     0     1     1     1     0     0     0     0     0     1     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     1     0     0     0     0     0     0     0     1     1     0     0
     0     0     1     1     1     0     0     0     1     0     0     1     0     0     0     0     0     0     0
     0     0     1     0     1     0     0     0     1     0     1     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1     0     0     0     1
     1     0     0     0     0     1     0     1     0     0     0     0     0     0     0     0     0     0     0
     0     0     1     0     0     0     0     0     0     0     0     0     1     0     0     0     1     0     1
     0     0     0     0     0     0     1     0     0     1     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     1     0     1     0     0     0     0     1     0     0     0     0
     0     0     1     0     0     0     1     0     0     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0     1     0     1     0     0     0     0
     0     1     0     0     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Column 20
```

Figure 3.1. Connection Matrix for estimation of error for PSO-GA algorithms.

```
particle_rate =

  Columns 1 through 11

   -0.0000   -0.0000    0.0000   -0.0000 -100.0000   -0.0000   -0.0000   -0.0000    0.0000    0.0000   -0.0000

  Columns 12 through 22

    0.0000   -0.0000   -0.0000    0.0000   -0.0000   -0.0000    0.0000    0.0000   -0.0000   -0.0000   -0.0000

  Columns 23 through 30

    0.0000    0.0000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -0.0000


particle_position =

       10000
```

Figure 3.2. Particle rate and particle position of each nodes in WSN.

```
Ffit =

   1.0e+004 *

   Columns 1 through 11

     7.5361    4.9397    4.9397    4.9397    4.9397    4.9397    4.4843    3.9077    3.5155    3.4874    3.2227

   Columns 12 through 22

     3.1346    2.8236    2.6711    2.6380    2.5853    2.5247    2.5230    2.4641    2.3552    2.2911    2.2270

   Columns 23 through 33

     2.1731    2.1173    2.0384    1.9423    1.8430    1.7392    1.6248    1.5433    1.5034    1.4846    1.4630

   Columns 34 through 44

     1.4109    1.3820    1.3530    1.3194    1.2983    1.2722    1.2548    1.2267    1.2086    1.1861    1.1707

   Columns 45 through 55

     1.1591    1.1521    1.1430    1.1394    1.1303    1.1240    1.1150    1.1076    1.1039    1.0968    1.0892

   Columns 56 through 66

     1.0826    1.0800    1.0714    1.0617    1.0549    1.0457    1.0396    1.0325    1.0320    1.0225    1.0225

   Columns 67 through 77

     1.0206    1.0206    1.0166    1.0166    1.0150    1.0145    1.0144    1.0134    1.0132    1.0119    1.0096
fx
```

Figure 3.3. Average analysis of FFit function.

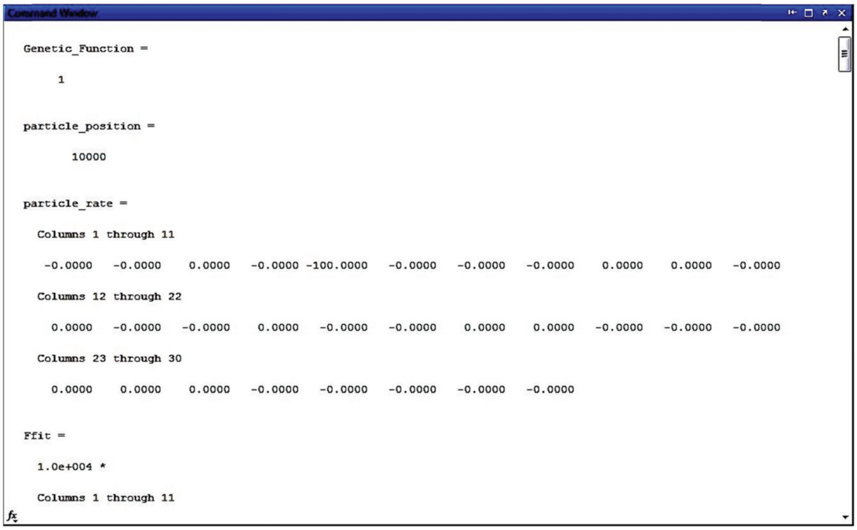

Figure 3.4. Implementation Evaluation PSO-GA algorithm.

(S-Server, N-Node).MAC identifier is one of the 48 bit identity which gives unique address to the mobile. Here in our proposed model we assign a local Temp id for each node in the network by combining MAC with the random number generated by the server.

48 bit address + random number = temp id

For example E8:65: AE: BF: 60:4E + 2584

= 241563 Temp id for Node 1.

3.3.2 Clustering

Clustering is defined as grouping of wireless sensor nodes, here we used K-Means clustering algorithm for clustering the sensor nodes. By using K-means clustering, a partition of sensor node of n observation is made into k-Clusters at which the observation is made by the nearest mean of the node which serves as the actual threshold of the centralized server.

3.3.3 Modified K-Means for clustering WSN

Observe interval data
Initialize K #—with one mean per cluster

```
For each interval data
          Assign the threshold value
Predict the mean of the neighbour nodes
Assign the centroid to the group of neighbour node
Move the centroid to the center based on mean and threshold
Activate the cluster
```

3.3.4 ACO with k-Means

```
FunctionACO_KMeans_Metadata
Initialize (cluster)
While (!grouped)
          Group_meta (nodes)
End while
          Threshold (nodes)
          Centroid (nodes)
          While (! termination)
CreateSolutions()
InspirationActions()
pheromoneUpdate()
end while
endFunction
```

3.3.5 BCO with k-Means

```
FunctionBCO_KMeans_Metadata
Initialize(cluster)
While(!grouped)
          Group_meta (nodes)
End while
          Threshold (nodes)
          Centroid ( nodes)
while (!termination)
          for i=1,…,ns
                    scout[i]=Initialise_scout()
                    flower_patch[i]=Initialise_flower_patch(scout[i])
                    do until stopping_condition=TRUE
                              Recruitment()
                              for i =1,…,nb
```

```
        flower_patch[i]=Local_search(flower_patch[i])
        flower_patch[i]=Site_abandonment(flower_patch[i])
        flower_patch[i]=Neighbourhood_shrinking(flower_patch[i])
        for i = nb,…,ns
        flower_patch[i]=Global_search(flower_patch[i])
        end while
end Function
```

3.3.6 Combination of hybrid model

```
FunctionACO_BCO_KMeans_Metadata
Initialize(cluster)
While(!grouped)
        Group_meta (nodes)
End while
        Threshold (nodes)
        Centroid (nodes)
while (!termination)
        for i=1,…,ns
        scout[i]=Initialise_scout()
        flower_patch[i]=Initialise_flower_patch(scout[i])
        do until stopping_condition=TRUE
        Recruitment()
        for i =1,…,nb
        flower_patch[i]=Local_search(flower_patch[i])
        flower_patch[i]=Site_abandonment(flower_patch[i])
        flower_patch[i]=Neighbourhood_shrinking(flower_patch[i])
                for i = nb,…,ns
                flower_patch[i]=Global_search(flower_patch[i])
                CreateSolutions()
                InspirationActions()
end while
end Function
```

3.3.7 Implementation Results

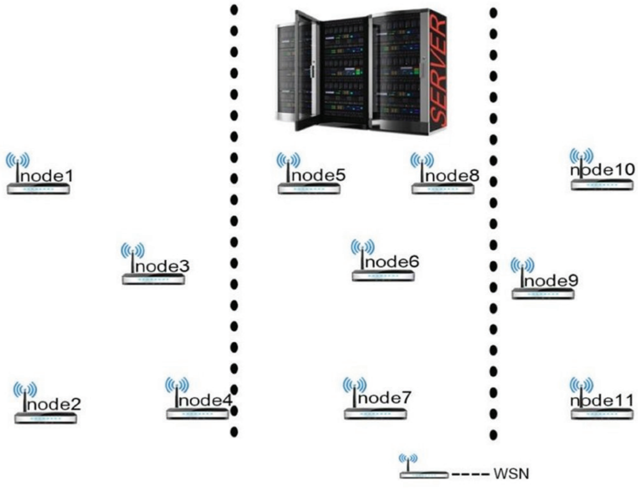

Figure 3.5. Sensor nodes with Server Conclusion.

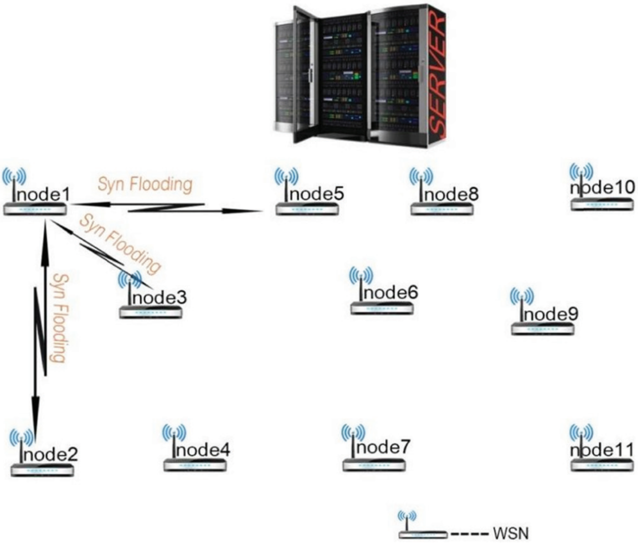

Figure 3.6. Sensor Node SYN flooding mechanism.

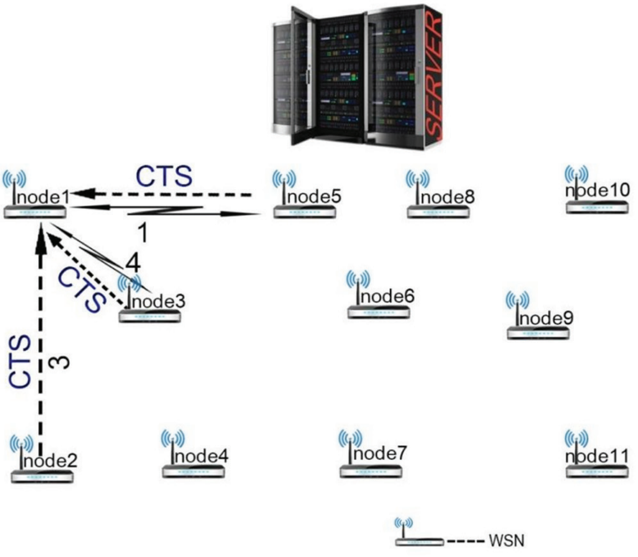

Figure 3.7. CTS message from sensor nodes.

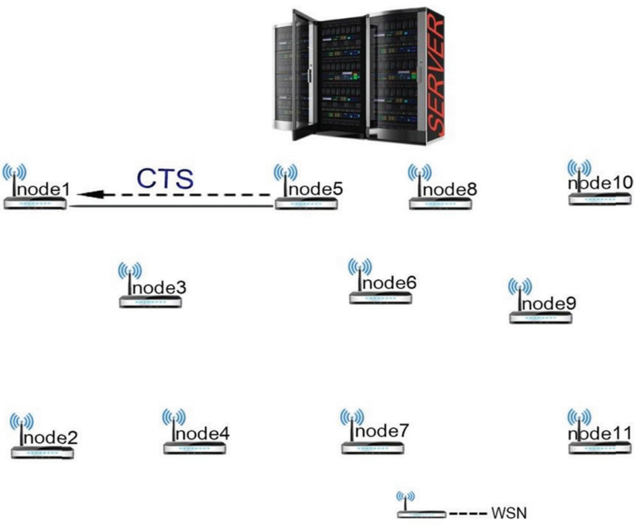

Figure 3.8. Optimal Path Establishment with Conformation CTS message.

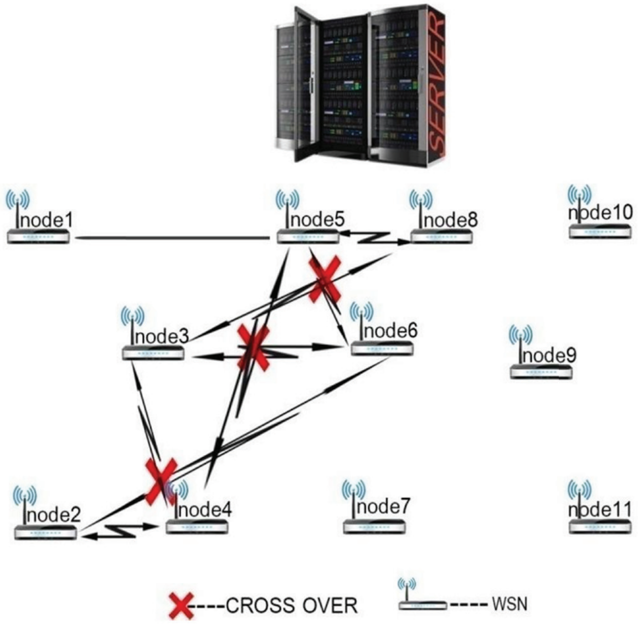

Figure 3.9. Crossover occurrence due to flooding of multiple RTS.

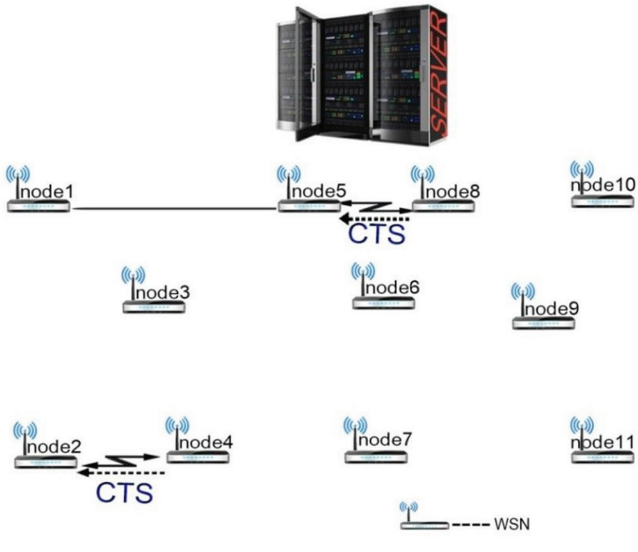

Figure 3.10. Node Discovery with Path Optimization.

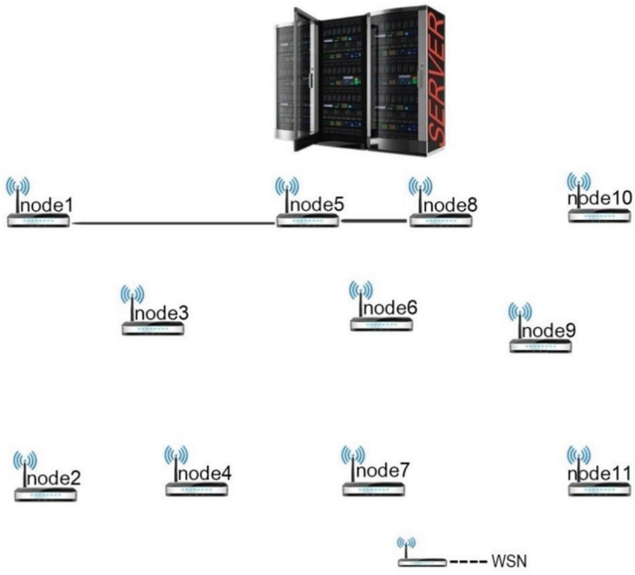

Figure 3.11. Path Establishment Between The Nodes.

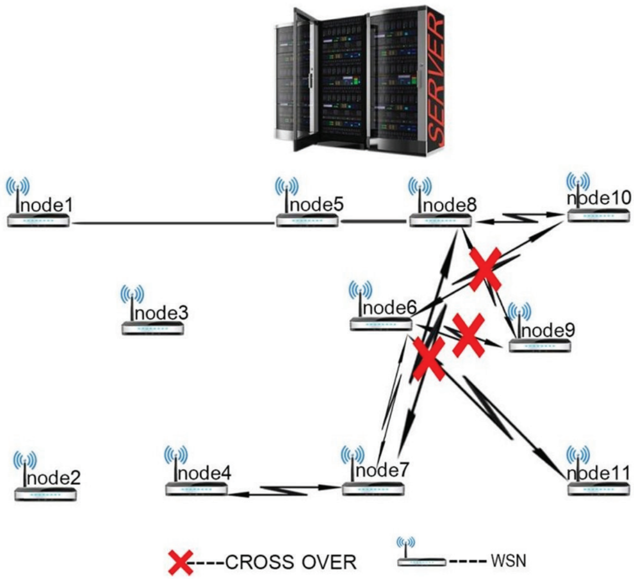

Figure 3.12. Node discovery for second level gateways.

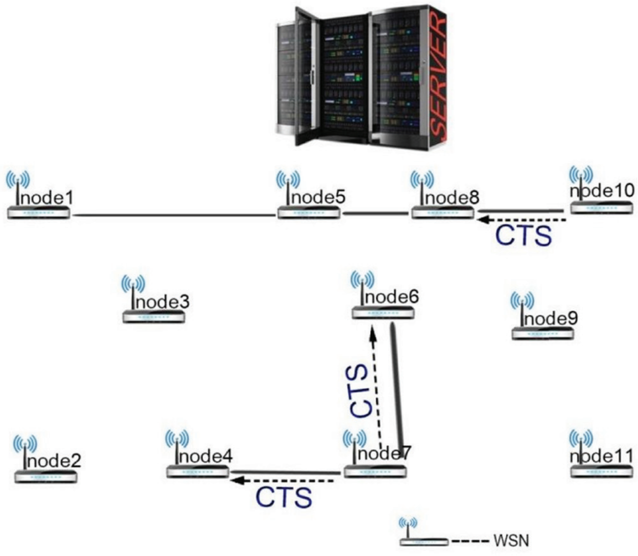

Figure 3.13. Decentralized node predictions (node 4, node 6, node 7).

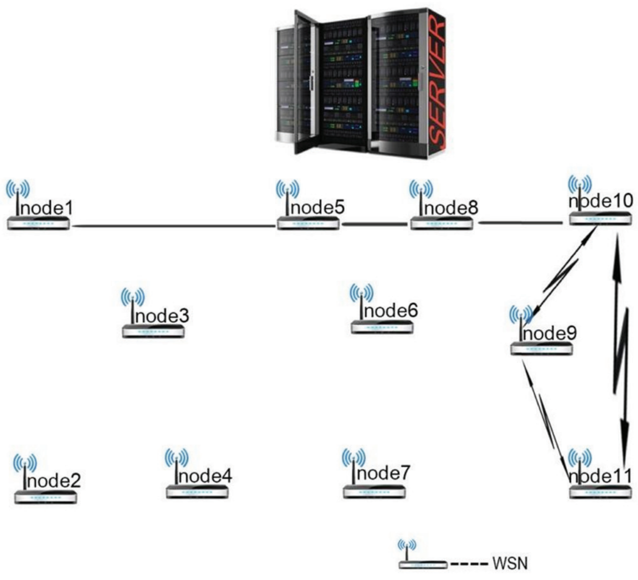

Figure 3.14. SYN flooding to discover the localized node (destination node).

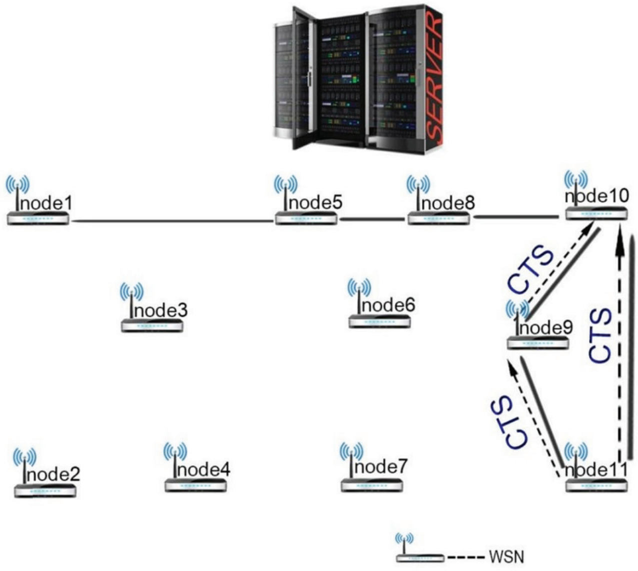

Figure 3.15. Repeated CTS message from the sensor nodes.

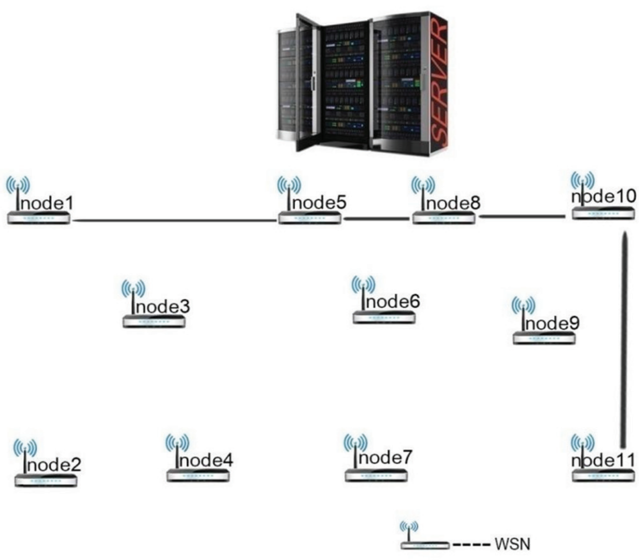

Figure 3.16. Optimal Path by node discovery.

3.4 PERFORMANCE ANALYSIS

3.4.1 Hybrid scheme 1

Table 3.1. The experimental result for Genetic, PSO, and Genetic-PSO.

Dimension	Genetic		PSO		Genetic-PSO	
	Worst case	Best case	Worst case	Best case	Worst case	Best case
40	-60	1.33e-5	0.7866	1.33e-60	0.9453	0.3458
60	-60	1.33e-5	0.7866	1.33e-6	0.9453	0.3458
80	-60	1.33e-5	0.7866	1.33e-6	0.9453	0.3458
100	-60	1.33e-5	0.7866	1.33e-6	0.9453	0.3458
120	8.7e-73	1.33e-5	-118	-120	-05e3.92	1.97e-106
140	8.7e-73	1.33e-5	-118	-120	-05e3.92	1.97e-106
160	8.7e-73	1.33e-5	-118	-120	-05e3.92	1.97e-106
180	8.7e-73	1.33e-5	-118	-120	-05e3.92	1.97e-106
200	8.7e-73	1.33e-5	-118	-120	-05e3.92	1.97e-106
220	2.0946	1.33e-5	3.8403	1.33e-60	-05e3.92	1.97e-106
240	2.0946	1.33e-5	3.8403	1.33e-6	-05e3.92	1.97e-106
260	2.0946	1.33e-5	3.8403	1.33e-6	3.54e-01	1.98e-41
280	-172	2.18e-10	3.8403	1.33e-6	3.54e-01	1.98e-41
300	-178	2.18e-10	3.8403	1.33e-60	3.54e-01	0.3458
320	-172	2.18e-10	3.8403	1.33e-6	3.54e-01	0.3458
340	-60	1.97e-41	3.8403	1.33e-6	3.54e-01	0.3458
360	-60	1.33e-5	2.18e-14	1.33e-6	3.54e-01	0.3458
380	-60	1.33e-5	2.18e-14	8.7e-73	6.40e-30	0.3458
400	-60	1.33e-5	2.18e-14	8.7e-73	6.40e-30	0.3458
420	-60	1.33e-5	2.18e-14	8.7e-73	6.40e-30	0.3458
440	-50	2.18e-10	6.8743	1.33e-60	6.40e-30	0.3458
460	2.0946	2.18e-10	6.8743	1.33e-6	6.40e-30	4.85e-07
480	2.0946	2.18e-10	6.8743	1.33e-6	8.7e-73	4.85e-07
500	2.0946	1.97e-41	2.18e-10	1.33e-6	8.7e-73	4.85e-07

Extensive simulation has been conducted to compare the performance of the proposed PSO localization algorithm to other energy-based source localization algorithms using Mat lab. Let the acoustic sensor and the node distribute in two dimensional square regions of size 100 by 100m, and the position of the source and the sensors are randomly chosen from within the sensor field in each run. We conducted 1000 repeated trials with equal intervals of 20 ts (timeslot), and the average value is localization error. All three energy-based acoustic source localization methods (Particle swarm optimization algorithm, multi-resolution searching (MR) and exhaustive searching (ES) algorithm) are used to calculate the source localization, and the error is recorded in each trial. The source energy is set at S=5000, c1 = c2 = 2 and 17 are random numbers uniformly distributed in [0, 1]. We conduct the trials with particle swarm optimization algorithm when the number of sensor in wireless sensor network is 20 and signal-noise-ratio (SNR) is 40db firstly. From the Figure 3.2, it is clear that the estimation error of three methods all decreases with increasing the number of sensor nodes in the wireless sensor network and the estimation error of PSO-GA is the least. The estimation error reduces rapidly while the nodes is more than 20, but when the number of network nodes reach 40 the estimation error doesn't change in evidence, therefore, increasing the nodes isn't meaningful for enhancing the localization precision and which will increase the burden of network.

Figure 3.17 denotes clearly PSO reaches a little estimation error with the increasing of SNR. Compared to other algorithms PSO has higher anti interference ability. However when SNR is large enough, it has little effect on improving proposed method performance. Therefore it is reasonable to set SNR to 30. This explains why the SNR is chosen respectively to be 30 and 50 in carrying out source localization. In this work, the computational complexity reflects the relationship of energy consumption indirectly because all of the three energy-based acoustic source localization methods are centralized algorithms with the same communication consumption.

In the simulation the size of the particle swarm is fixed at 20, multi-resolution searching style is 4*4+25*25 and the step of exhaustive search method is 2, and the number of iteration is (100/2)* (100/2). The iteration of three algorithms is shown in Table 3.1.

The Table 3.2 derived from Table 3.1, it is noted that the computational complexity of PSO algorithm is lowest and it has the highest localization precision with the same parameters in same trail. For MR and exhaustive search algorithms, enhancing localization precision must be at the cost of increasing computational complexity because of them being the computational method based on iteration. Compared to other

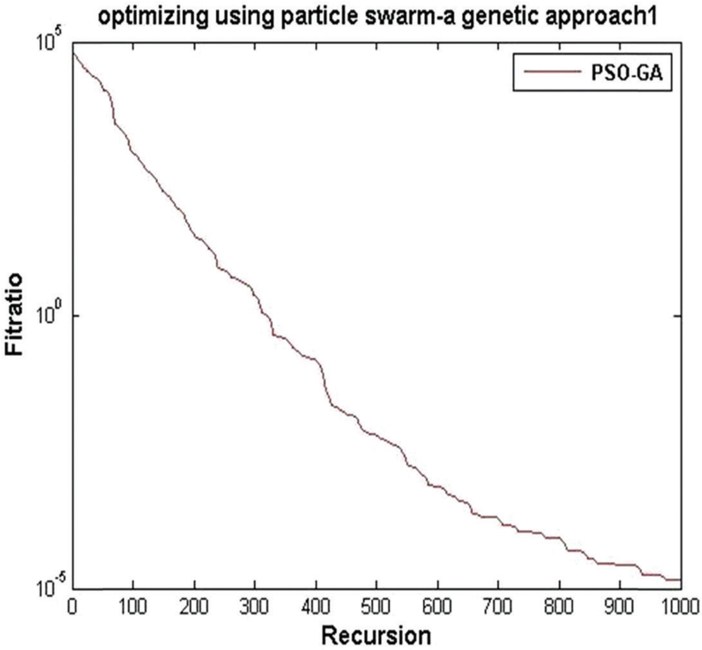

Figure 3.17. Results of PSO-GA algorithm.

Table 3.2. Computational complexity of PSO algorithm.

Complexity	PSO	MR	Exhaustive Research
Recursion ratio	400	641	2500

conventional optimization methods PSO has lots of advantages such as it hasn't special requirements at the form of objective function which makes it have extensive application. At the same time based on the characteristics of evolution for the probability ensures the rapidity of the method, so for the optimization of complex computational problem PSO has a strong advantage.

3.4.2 Hybrid Scheme 2

The simulation-experimental setup is tested in OMNET++ with 11 sensor nodes and with one server node. Figure 3.5 clearly states the experimental setup. Figure 3.6 states the SYN flooding mechanism of RTS message sent from a sensor to discover neighbour nodes. Once the SYN flooding discovers the neighbour, the K-Means cluster is initiated with the mean value, the centroid prediction and cluster head assignment

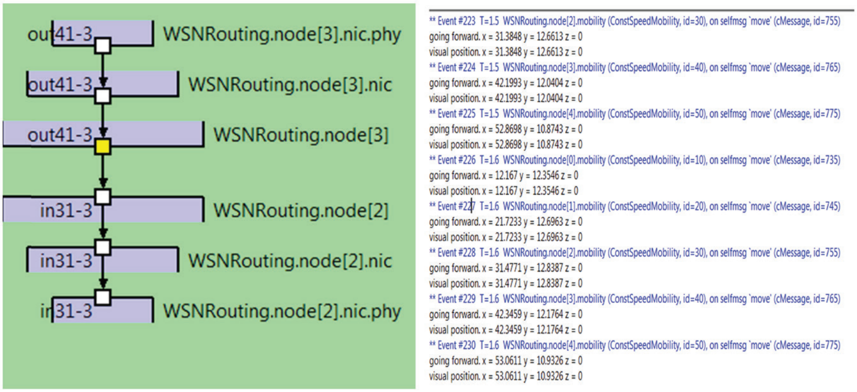

Figure 3.18. The packet transmission within the cluster with repeated sequence of CTS.

in sensor node is not achieved, but still the problem has better solution than the approached one. Once the CTS message denoted in Figure 3.7 is received from the neighbour nodes, the node is discovered and optimal path denoted in Figure 3.7 is established between nodes. Figure 3.8 states that during the node discovery, multiple nodes communicate to find the nearby neighbour node. Figures 3.8–3.11 denotes the scenario of ACO and BCO, node 5 to node 8 is achieved using ACO and node 2 to node 4 is achieved using BCO, since node 4 is already discovered and has largest repeated threshold, hence the node 4 is eliminated.

3.5 CONCLUSION

Here a new hybrid scheme of data routing has been proposed using PSO-GA and PSO–K-means clustering. The localizations are optimized by using the presented method and have been evaluated, validated with extensive simulation study which consistently promises superior performance and is easy to implement as compared with MR and exhaustive searching localization methods. The hybrid scheme addressed here yields high result when the node availability is centralized but when it is clustered and centralized (Figure 3.14 and Figure 3.18), the proposed approach is performing better with 60% accuracy level. Since the node clustering in WSN is not in wider range, during the node failure stated in Figure 3.13), K-Means algorithm fails to find the cluster Figure 3.15. From the study, the results have demonstrated that the proposed approach has higher precision and lower computational complexity in acoustic source localization for the wireless sensor network.