Data Gathering Scheme for Mobile Wireless Sensor Network with Contention Protocol

Jayanth C[1], Vinay Prasad M.S[2]

[1]Dept.of Electronics and communication, Sri Jayachamarajendra collage of Engineering, Mysuru, India
[2]Professor, Dept. of Electronics and communication, Sri Jayachamarajendra collage of Engineering, Mysuru, India

---

Abstract - *Data acquisition is an important operation in various mobile wireless sensor networks (MWSN) applications. The mobile sink is used to solve the problem of nodes near the sink running out of energy as a result of the massive volume of data being transmitted. The use of the Contention protocol has helped to reduce collisions. The protocol not only avoids network collisions, but also stops two nodes from transmitting broadcasts at the same time. In effect, the suggested data gathering with collision control for Mobile Sinks is based on a bee colony. The suggested approach is compared to the present method, and parameters such as throughput, latency, and packet delivery ratio are analyzed using the NS 2 simulation tool.*

Key Words: **Congestion mitigation, Contention window, Artificial Bee colony Algorithm.**

1.INTRODUCTION

In recent years, wireless sensor networks (WSNs) have been used for a variety of purposes, including animal surveillance, vehicle tracking, weather forecasting, network monitoring, medical services, seismic detection, and more [1]. Data acquisition is one of the important operations utilised in WSN, and as a result, it has attracted a lot of interest from professionals and academics [2]. The Wireless Sensor Network is composed of thousands or more number nodes that can be deployed at random or in a fixed location to monitor the surroundings. WSNs have been a popular trend in recent years, and they necessitate the deployment of a large number of tiny nodes. Changes in the environment are then detected by the nodes and sent to other nodes via a versatile network environment. Sensor nodes are appropriate for use in harsh environments or over long distances. Generally, in a typical data collection application, all nodes are typically set in place to collect data before being sent to the Sink via the routing protocol. The energy hole problem is created when data streams work in a many to one fashion, which leads to nodes near the mobile sink gets huge traffic loads, causing premature energy depletion and the formation of an energy hole around sink, is one of the most difficult unresolved difficulties with this process. The energy of sensor nodes is restricted, resulting in a communication overhead problem. The data from some nodes cannot be transferred because the network is not connected leads to the communication constraint problem hence it's necessary to employ reliable strategies when the network is not connected for communication reliability. Congestion is a situation in which data packets are lost, the network is delayed, and the network's performance is degraded. Node mobility has been used by previous studies to solve the issues mentioned above. The mobile node serves as a data collection agent, travelling through the network along predetermined paths to data acquisition in the monitoring area. By utilising mobile node properties such as coverage, connection, and energy distribution, MWSNs are typically built dynamically or updated with real-time conditions to fill in routing voids within the network and blind zones inside the sensor.

Previous research has also suggested a set of data collecting algorithms based on solutions for mobile Sinks [3–4] [9– 10], which have partially overcome the three challenges highlighted. Existing approaches, on the other hand, struggle to account for mobile path length minimization, data collection maximisation, and network reliability optimization all at the same time.

We concentrated on the MWSN with congestion control in this work. The Sink chooses its moving path dynamically, such that the next-hop location is computed and determined before the next data capture, based on network environment factors and moving strategy; While the cluster member nodes send data to the cluster head node for temporary storage, the artificial bee colony optimization is used at the same time to find the shortest path to each cluster head node. The cluster head node transfers the stored data to the mobile Sink after the mobile Sink has arrived at the cluster head node. The saved data is sent to the database using Mobile Sink. We present a metaheuristic algorithm based on this data acquisition programme that takes into account cluster head selection, the path from cluster member to the cluster head, and the mobile Sink path that supports the artificial bee colony. The suggested system can successfully minimise network congestion, data transmission, while also increasing network reliability and data gathering efficiency and extending the network's lifetime.

The following is how the rest of the paper is organized: The application scenario analysis and problem description are covered in Section 2. The" ABC algorithm", as well as the mathematical equations and optimization techniques used for MWSN data gathering and congestion control are discussed in section 3. Section 4 contains the parameter setting and simulation results that validate the proposed system's performance, and Section 5 conclusion of the paper.

2. PROBLEM DESCRIPTION AND APPLICATION SCENARIO

The major goal of this study was to create a mobile sink for data acquisition that used a heuristic artificial bee colony method with congestion control to handle problems like data collection capacity, network efficiency, and packet delivery ratio.

Three particular optimization goals were included in this:

1) Accelerate data acquisition quantity.

2) the mobile Sink's shortest travelling path.

3) Optimize network efficiency by collecting as much data as possible.

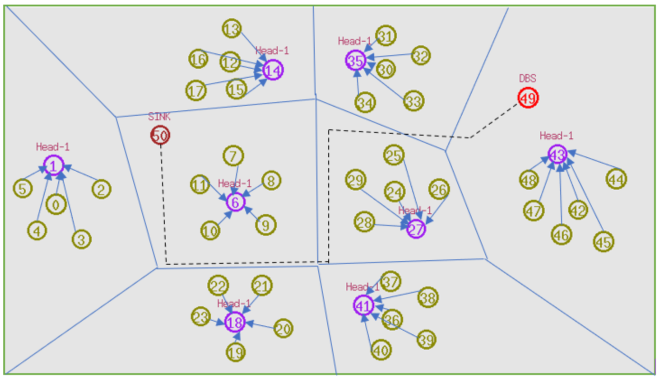

Fig -1: Application Scenario

The application scenario had following characters

1) Cluster heads are chosen depending on the node with the maximum residual energy.

2) Cluster members are unmovable. It transmits data packets to CH and shortest possible path info.

3) The Sink nodes can move free and collects data packets from CH. The collected data packets transmit to database server.

4) The sink travel in the shortest traversal path. [11-14]

Input: list of non-elected nodes
Output: list of clusters' members

```
Begin
  if (CHs are elected) then
    for each node i ∈ n do
      if (node is non-elected node and energy > 0) then
        compute distance between node i and MS
        for each node j ∈ CHs do
          compute the distance between nodes i and j
          compare the result with distance to MS
          join to the closest one
        end for
        transmit packet to the closest node j
      end if
    end for
  end if
end
```

Fig – 2: Cluster formation Pseudo code

3. ABC ALGORITHM AND CONGESTION CONTROL

3.1 Artificial Bee Colony Algorithm

Artificial bee colony method [7] is a quick, efficient, simple, and easy-to implement optimization algorithm characterised by distributed computation and high speed. There are three "bees" groups in the "colony": onlookers, scouts, and employed bees. Bees are used to finds the optimal path. The nectar amount of a food source corresponds to the fitness of the solution. There are two groups of Unemployed bees: Onlooker bees and scouts. An onlooker bee chooses a food source on the dance area; a food source may or may not generate a new solution that depends on the random number selected for a particular onlooker bee as well as the probabilities of the food source. Scouts' bees choose their food sources randomly. Employed bees look for new food sources having more nectar within the neighborhood of the food source.

There are four main steps in the ABC algorithm [5].

1) Initialization phase: Ss is the swarm size, where swarm size is nothing but population size. Initialize the randomly distributed initial population of solution Ss. Let

$$yi = \{yi, 1, yi, 2........yi, n\}$$

Where i solution in the swarm, n is the dimension size. The initial population is then

$$yi = ymin + random\ (0, 1) * (ymax - ymin)$$

2) Population Updating phase: The initial new solutions (positions of food sources) are generated at random. A food

source was assigned to each employed bee. Position for a new food source Using the equation

$$V_{ij} = y_{ij} + random\ (y_{ij} - y_{kj}),$$

calculate Vij in the neighbourhood of yij for the employing bees. Where k is a solution in the neighbourhood of i and rand is a random number between -1 and 1. In each iteration, calculate the amount of nectar in the new food source and compare it to the old one using the greedy selection procedure. If the new source's nectar yield is larger, the employed bee will switch to it; otherwise, it will stay with the old one.

3) Bee Source Selection: Determine the probabilities values. Pi for the solutions yi based on their fitness values using the equation

$$Pi = \frac{fit}{\sum_{i=1}^{Ss} fit\ i}$$

The following equation was used to compute the fitness values of solutions.

$$fit\ i = \begin{cases} \dfrac{1}{1+fi}, & fi \geq 0 \\ 1 + abs(fi), & fi < 0 \end{cases}$$

4) Population Elimination: If a position can't be improved after a certain number of cycles (called a limit), find the abandoned solution (source) and replace it with a new randomly generated solution Xi for the scout using the equation

$$y_{ij} = ymin_j + random\ (0, 1) * (ymax_j - ymin_j).$$

Keep in mind the best food source location (solution) you've managed to acquire so far.

The ABC method is a high-speed, distributed computing optimization technique that is efficient, simple, intelligent, and straightforward to implement.

The Sink completes the complex centralised optimum algorithm in the data collection communication protocol, which is of two stages: the initialization and the data collecting phase, given that the mobile Sink's storage, energy, and algorithm sources are not constrained.

The first step is the initialization phase, during which the entire network topology is gathered, after which cluster head nodes are chosen and cluster member nodes are added to each cluster head (figure 1 and 2). By the end of this procedure, all nodes had received the cluster head's shortest hop information and had transmitted the relevant cluster head information. The data is then sent to the mobile sink with the shortest hop in the next stage. Cluster nodes keep track of the first and last times they enter and exit the mobile Sink's communication range. The mobile Sink assigns communication time to each cluster head based on this information. The mobile Sink then broadcasts the results of the previous calculation to the network, which leads to series of matching lists between cluster member nodes and cluster heads in the next phase. When each node receiving this broadcasting information obtains its target cluster head information, eliminates relevant information from the broadcast information, and continues broadcasting, the ideal cluster head selection process for the whole network is complete.

The data collection phase begins after that. All network nodes continue to collect data after initialization and send it to their assigned cluster head via the defined route tree. Each cluster head saves itself as well as the sensory information of other member nodes before the mobile Sink arrives. Before collecting data for the next round, the sink selects its mobile path dynamically. It calculates and selects the next hop's objective location based on the present network environment's factors and mobile strategy for determining the shortest route to each cluster head and node.

3.2 Congestion control

The carrier sense multiple access/collision avoidance (CSMA/CA) was used to reduce the likelihood of a collision. The CSMA/CA protocol not only avoids network collisions, but also stops two nodes from delivering broadcasts at the same time. Every node retains the neighbour table once the broadcast messages have been exchanged. The number of neighbour nodes, their initial energy, and their distance from the mobile sink or each other are all recorded in the neighbour table [8]. Congestion mitigation is accomplished by adjusting the current node's CW in the following manner:

$$CW = (CWmin + 1) * 2^n - 1 \quad .... (1)$$

Here, the adjusted initial contention window of the current node is denoted by CW. CWmin represents the contention window's minimum, and n represents the window adjustment parameter, which is given by:

$$n = ((qnext-t)/(Qmax-t)) \times \log_2((CWmax+1)/(CWmin+1)) \quad .... (2)$$

where qnext is the next hop node's queue length, t is the congestion threshold, Qmax is the node buffer queue's maximum length, and CWmax is the maximum contention window The adjustment range of the CW, according to Equations (1) and (2), is between CW min and CW max. When qnext=t, n=0 and CW=CWmin. As the length of the node queue grows longer, the CW gets larger, and the priority of the node to access the channel decreases. When qnext = Qmax, n is given by:

$$n = \log_2((CWmax + 1)/ (CWmin + 1)) \quad .... (3)$$

The value of CW is represented as CWmax from equation (3) into Equation (1), indicating that the current node has the lowest priority to access the channel. The current node waits the longest duration, allowing following hop nodes adequate time to send packets from the buffer queue.

Short inter-frame spacing [SIFS] is the smallest inter-frame space that is used to separate transmissions pertaining to a single conversation. There is only one station transmitting at any given moment, giving it priority over all other stations. This value is constant for each PHY and is calculated so that the transmitting station can switch back to receive mode and decode the incoming packet. Point Co-ordinate inter frame space [PIFS] is the access point uses this to get first access to the media before any other station. It is calculated as:

P IF S = SIFS + 1stslottime

Distributed inter frame space [DIFS] is the inter frame space available to a station that wants to start a fresh transmission. It is calculated as:

DIF S = SIFS + 2ndslottime

Slot Time= Time required to detect a frame + Propagation delay +Time required to transition from Rx to Tx state+ It's time to inform the MAC layer of the channel's status.

Table 1 shows the set of parameters used to Control the congestion.

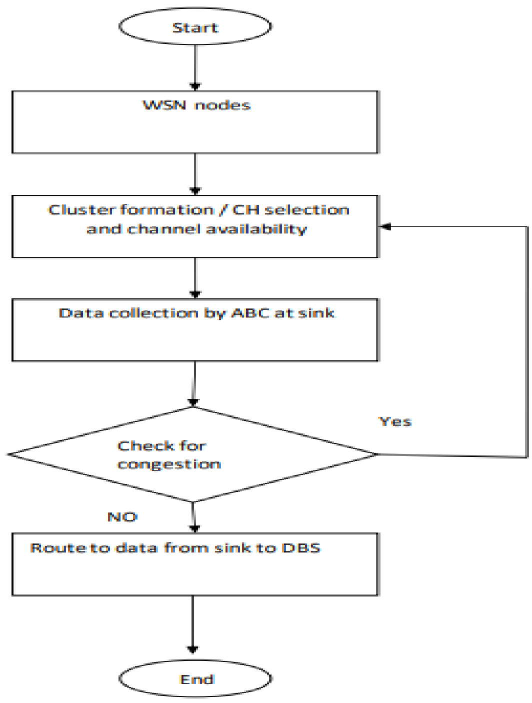

Fig -3: Data acquisition using ABC algorithm at mobile sink with Congestion control flowchart.

Table -1: Typical Parameters Values

Parameters	**Value**
CWmin	15
CWmax	1023
Slot Time	9 µ sec
CCATime	3 µ sec
Slot Time	9 µ sec
RxTxTurnaroundTime	2 µsec
Frame Space Time	16 µsec
Distributed Inter Frame Space	28 µ sec
PreambleLength	96 bits
PLCPHeaderLength	40 bits
Shortest Inter PLCPDataRate	6 Mbps
Frequency	2.4 GHz
MaxPropogationDelay	0.5 µsec

4. SIMULATION RESULTS AND ANALYSIS

4.1 Simulation Environment

The proposed method's performance was evaluated using the NS-2 network simulator. 50 nodes and a mobile Sink were employed equally at random in a 1300*1000 m2 square area in our simulated scenarios.[15]; other parameter setting is listed in Table 2

Table -2: SIMULATION PARAMETERS

Simulation Parameters	**Value**
Network size	1300*1000
No of nodes	50
Mac layer	802.11
Simulation time	40 sec
Traffic	CBR
Initial energy	100 J
Antenna model	Omni Antenna
Radio Propagation model	Two ray grounds
Phy	Phy/Wirelessphy
Packet Size	512 bytes

4.2 Performance Analysis

1) Throughput: Every second, the ratio of packets can be transferred from source to destination. Throughput can be given as:

$$\text{Throughput} = x1/x2 * 100$$

Throughput is measured in kbps, where x1 is the number of packets received at the destination and x2 is the number of transmitted packets from the source. When congestion is reduced, the time a data packet spends waiting in the queue is reduced, ensuring that the data packet sent reaches its destination as promptly as possible. The network's throughput has grown as a result. ABC with congestion control is better than without congestion control. ABC with congestion control is about 0.13% higher than that. The graph depicts the throughput.

2) Packet Delivery Ratio: PDR is the percentage of total data packets created divided by total received packets. Packet delivery ratio can be given as:

$$\text{PDR} = nr/ns$$

where nr denotes the number of received data packets and ns denotes the number of data packets transmitted. The packet delivery graph is shown in the image, and it can be seen that packet processing is dependent on the intake and outflow of packets inside the queue. The MAC layer flag shows the congestion condition during the packet process by exchanging RTS and CTS messages. ABC with congestion control delivered 20% more packets to their destination than ABC without congestion control. In our suggested method, congestion is managed such that packets are processed effectively, resulting in more packets reaching their destination.

3) Delay: Latency is the average difference in time between the source and destination nodes. It's calculated by subtracting the packet sent time from the packet arrival time. The average end-to-end delay can be computed as:

$$\text{Latency} = 1/Nr \sum (r - s)$$

Where r is the reception time at the sink node, s is the Data transmission time at the source node, Nr denotes the total number of packets that have been successfully received. The delay graph is shown in the figure. ABC with congestion control is about 73% lesser delay than ABC without congestion control. It has been noted that the proposed system's delay is significantly less than that of the existing system.

4) Overhead: The number of routing packets sent per the data packet has arrived at its destination. A routing packet's hop-by-hop transmission is counted as one transmission. The routing overhead specifies how many routing packets are required to propagate data packets for route discovery and maintenance. It is reduced by 12% when compared with ABC without congestion control. When Congestion is controlled number of routing packets is less.

$$\text{Overhead} = \text{RoutingPacketsCount}$$

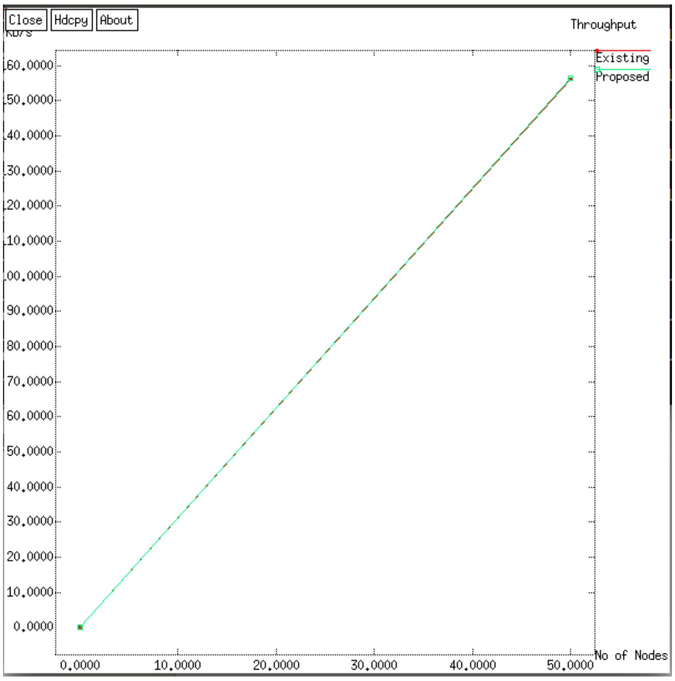

Fig -4: Throughput in kbps

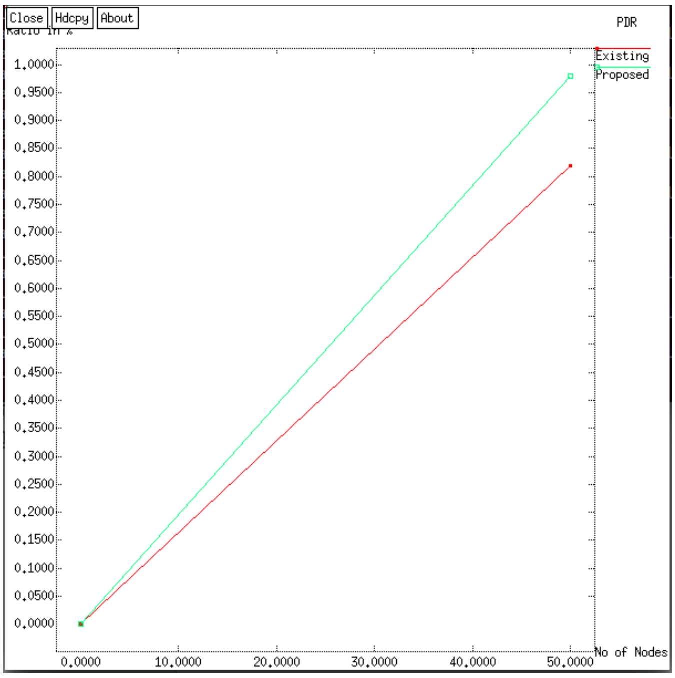

Fig -5: Packet Delivery Ratio

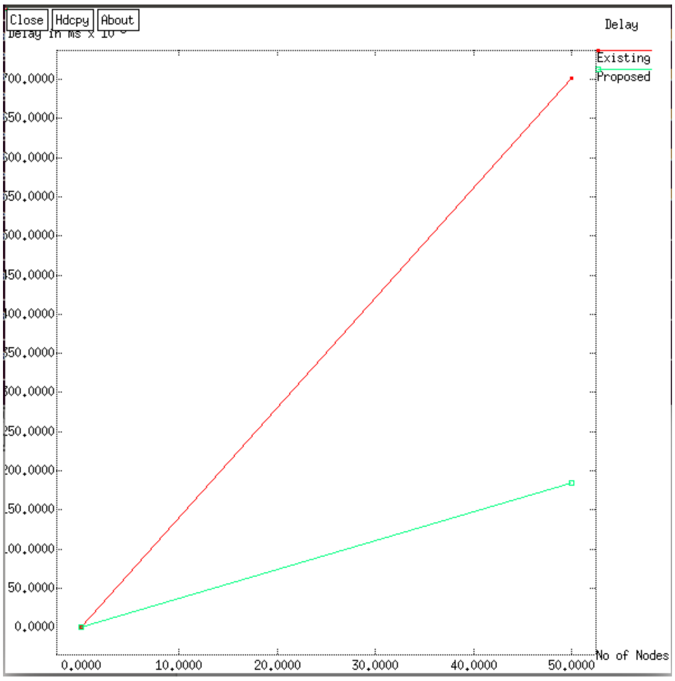

Fig -6: Delay

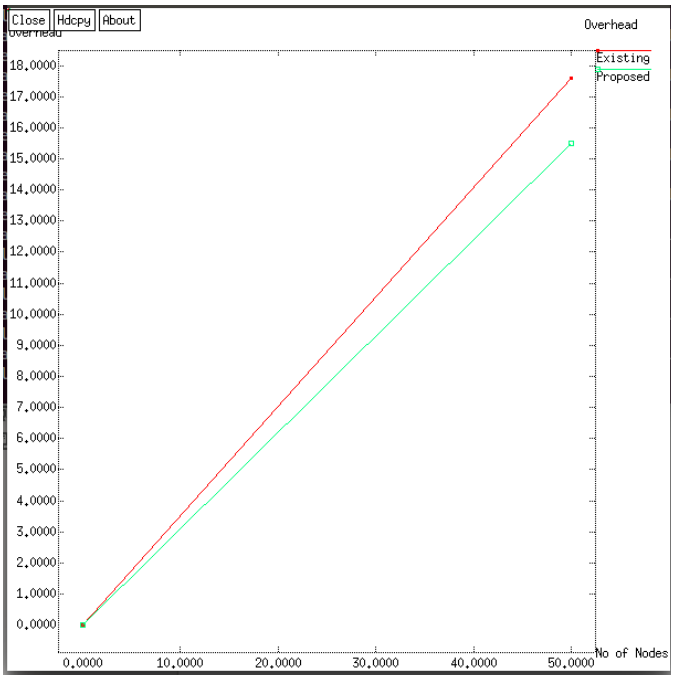

Fig -7: Overhead

5. CONCLUSIONS

The usage of CSMA/CA to control network collision was investigated in this study. With the help of a heuristic artificial bee colony, we were able to collect data using a mobile sink. Simulation outcomes show that the proposed work enhances WSN throughput, PDR, end-to-end delay, data collection efficiency, and overhead when compared to equivalent existing work.

REFERENCES

[1] D. Mascarenas, E. Flynn, C. Farrar, G. Park, and M. Todd, "A mobile host approach for wireless powering and interrogation of structural health monitoring sensor networks," IEEE Sensors Journal, vol. 9, no.12, pp. 1719–1726, 2009.

[2] G. Xing, M. Li, T. Wang, W. Jia, and J. Huang, "Efficient rendezvous algorithms for mobility-enabled wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 11, no. 1, pp. 47–60, 2012.

[3] S. Park, E. Lee, M.-S. Jin, and S.-H. Kim, "Novel strategy for data dissemination to mobile sink groups in wireless sensor networks," IEEE Communications Letters, vol. 14, no. 3, pp. 202– 204, 2010.

[4] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," IEEE Transactions on Automatic Control, vol. 56, no. 10, pp. 2445–2455, 2011.

[5] W. Gao, B. Zhao, G. T. Zhou, Q. Y. Wang, and C. Y. Yu, "Improved Artificial Bee Colony algorithm based gravity matching navigation method," Sensors, vol. 14, no. 7, pp. 12968– 12989, 2014.

[6] Kennedy J, Eberhart R C. Particle swarm optimization[A]. In: Proceedings of IEEE International conference on Neural Networks[C]. Perth, Australia: [s. n.], 1995. 1942- 1948.

[7] X. Zhang, X. Zhang, S. Y. Yuen, S. L. Ho, and W. N. Fu, "An improved artificial bee colony algorithm for optimal design of electromagnetic devices," IEEE Transactions on Magnetics, vol. 49, no. 8, pp. 4811–4816, 2013.

[8] Vuran, M.C.; Akyildiz, I.F. XLP: A cross-layer protocol for efficient communication in wireless sensor networks. IEEE Trans. Mob. Comput.2010, 9, 1578–1591.

[9] T.-C. Chen, T.-S. Chen, and P.-W. Wu, "On data collection using mobile robot in wireless sensor networks," IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 41, no. 6,pp. 1213–1224, 2011.

[10] M. Zhao, J. Li, and Y. Yang, "A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks," IEEE Transactions on Mobile Computing, vol. 13, no. 12, pp. 2689–2705, 2014.

[11] M. Zhong and C. G. Cassandras, "Distributed coverage control and data collection with mobile sensor networks," IEEE Transactions on Automatic Control, vol. 56, no. 10, pp. 2445–2455, 2011.

[12] T.-C. Chen, T.-S. Chen, and P.-W. Wu, “On data collection using mobile robot in wireless sensor networks,” IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, vol. 41, no. 6, pp. 1213–1224, 2011.

[13] M. Zhao, J. Li, and Y. Yang, “A framework of joint mobile energy replenishment and data gathering in wireless rechargeable sensor networks,” IEEE Transactions.

[14] M. Ma, Y. Yang, and M. Zhao, “Tour planning for mobile data gathering mechanisms in wireless sensor networks,” IEEE Transactions on Vehicular Technology, vol. 62, no. 4, pp. 1472– 1483, 2013.

[15] J. Wang, J. Cho, S. Lee, K.-C. Chen, and Y.-K. Lee, “Hop-based energy aware routing algorithm for Wireless sensor networks,” IEICE Transactions on Communications, vol. 93, no. 2, pp. 305– 316, 2010.