Chapter 5

Obstacle-avoiding data routing scheme for IoT-enabled WSNs

5.1 Introduction

The Industrial Internet of Things (IIoT) performs significant transformations in the relationship between humans and the operation of physical devices [119, 27]. IIoT is used in broad areas of applications such as the agriculture industry, healthcare industry, manufacturing industry, and chemical industry [120, 121, 122]. One of the prominent applications of IIoT is industrial safety and security, where critical infrastructures such as pipelines, power plants, and machine conditions are monitored through IIoT-based systems [123, 124]. Failure of industrial infrastructure and machinery releases hazardous substances such as harmful gasses and liquids within the environment [125]. It causes huge economic losses and harms the physical health of workers [126]. Various sensors, such as vibration, humidity, temperature, and gas sensors are used to monitor infrastructure and machine health conditions [127]. These sensors constitute Wireless Sensor Networks (WSNs) that assist IIoT applications in detecting and alerting early infrastructure and machine failures. It significantly reduces economic losses and saves the physical health of the workers. In WSNs, sensor nodes are equipped with lim-

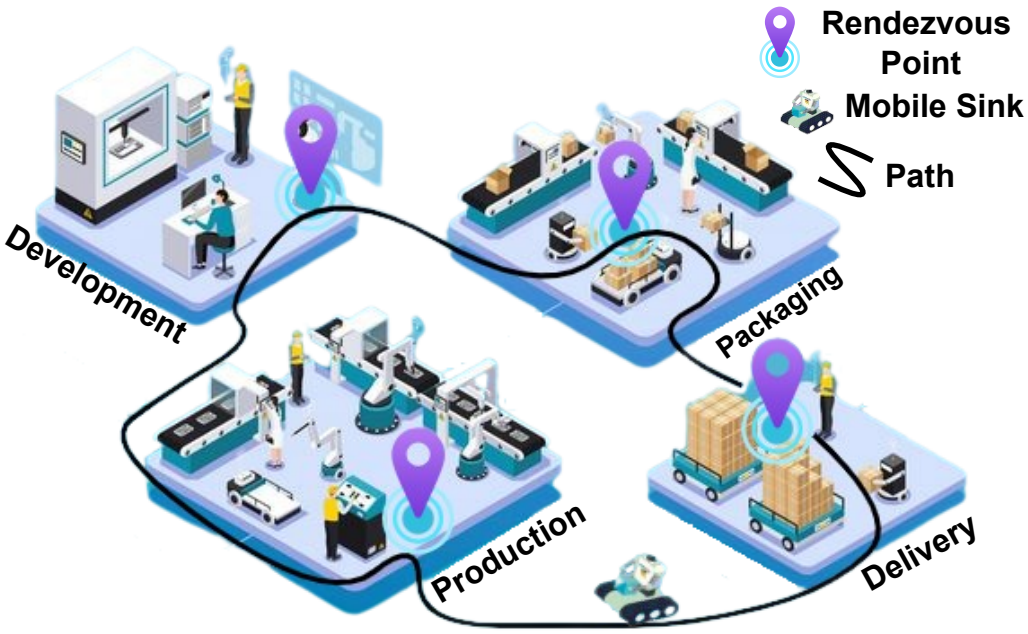


Figure 5.1: Obstacle-aware data collection using MS in an industry.

ited batteries [128]. Therefore, energy consumption minimization at sensor nodes is one of the major requirements to increase the lifetime of IIoT-enabled WSNs [129]. Nowadays, a Mobile Sink (MS) based data collection mechanism is very popular for energy-efficient data gathering from the sensor nodes [130]. In MS-based data collection, a set of Rendezvous Points (RPs) are selected for data gathering. RPs are data collection points/locations where MS stops for data gathering from the sensor nodes [131]. However, the presence of physical obstacles within the network prevents MS movement and creates transmission faults within the networks [132]. It drastically reduces overall network performance and Quality of Services (QoS). Industrial safety and security applications need high-quality data with minimum delay. It helps to trigger alarms to notify relevant authorities about the hazard and extinguish the fire/harmful gasses before it spreads extensively. Fig. 5.1 shows MS-based data collection in an industrial environment.

Recently, some MS-based obstacle-aware data-gathering schemes for WSNs have

been proposed [39, 40, 75]. In these approaches, the designed paths for MS movement are too close to obstacles [72, 73], which creates a risk of collision between MS and obstacles during the MS movement. It significantly reduces QoS and increases data collection delay. Furthermore, MS movement paths are not smooth and have sharp turns at intersections. So, MS must decelerate abruptly during movement. It hampers data collection and increases MS travel time. Existing approaches do not optimize the load among sensor nodes [71, 72]. It causes high energy consumption at nodes and leads to poor network lifetime. Furthermore, existing algorithms are very difficult to implement in resource-constrained WSNs due to their complex design [133, 67]. These issues motivate us to propose an obstacle-aware data collection scheme for IIoT-enabled WSNs.

The proposed scheme uses the Manta-Ray Foraging Optimization (MRFO) algorithm to identify the optimal number of RPs within the network. It significantly minimizes the energy expenditure of sensor nodes and improves the overall network lifetime. The MRFO has a lesser computational cost, greater accuracy, and higher performance as compared to other meta-heuristic algorithms [134]. Therefore, it takes minimum resources during the execution. Furthermore, the EBS-A* algorithm is used to design a smooth and collision-free optimal obstacle-avoiding path for MS [135]. It reduces the time complexity by half as compared to the traditional A* algorithm using a bidirectional search. The safety of the designed path is evaluated using a collision avoidance function [136]. The simulation and testbed results illustrate that the proposed scheme outperforms as compared to the state-of-the-art algorithms. The main contributions of this chapter are as follows.

- This chapter proposes a novel MRFO-based clustering algorithm that identifies the optimal number of RPs and forms optimal clusters within the network. It significantly reduces the energy expenditure of sensor nodes and improves overall network lifetime.

- This chapter proposes an optimal obstacle-aware path planning scheme using the EBS-A* algorithm. The proposed approach removes sharp turns and edges that enable smooth MS movement. The proposed approach also minimizes the MS path length and data collection delay.

- The proposed obstacle-aware data routing scheme is also applied to monitor the industrial infrastructure and machine conditions in a chemical factory. The proposed scheme is applicable in resource-constrained WSNs.

- Extensive simulations and testbed results show the effectiveness of the proposed MS-based obstacle avoidance data routing scheme in terms of network lifetime, stability period, residual energy, data collection delay, and MS safety assessment.

Table 5.1: Terminologies and definition

Terminologies	***Definition***
$\mathcal{N}$	Number of deployed sensor nodes.
$\mathbb{S}$	Set of sensor nodes. $\mathbb{S} = s_1, s_2, s_3, s_4.....s_{\mathcal{N}}$.
m	Number of obstacles.
p	Total number of rendezvous points in WSN.
$\mathbb{R}$	Rendezvous points set. $\mathbb{R} = \mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3....\mathcal{R}_p$.
$\mathcal{R}_\mathcal{B}$	Base station.
rc	Sensor nodes' communication range.
$\mathcal{T}_{\mathcal{D}ist}(s_i)$	Euclidean distance from s_i to the next-hop node.
$\hat{\mathrm{D}}(s_i, s_j)$	Euclidean distance from node s_i to s_j.
$\hat{\mathrm{H}}(s_i)$	Number of hops between s_i and RP.
$\mathcal{N}_{\mathcal{CN}}(s_i)$	Number of child nodes of the sensor node s_i. Child nodes are the nodes whose data is forwarded to RP by node s_i.
$allot(s_i)$	It contains the id of the RP to which the sensor node s_i is assigned.
$\mathbb{L}$	Length of the MS data collection path.
C_s	Current grid cell in forward search direction.
C_d	Current grid cell in backward search direction.
S_g	Source grid cell.
D_g	Destination grid cell.
$NG_i(C_s)$	Neighbour grid cells of cell C_s.

5.2 System Model and Application Scenario

In this chapter, $\mathcal{N}$ number of sensor nodes are randomly deployed in the $\mathcal{D}$ X $\mathcal{D}$ $[m^2]$ area. The BS is aware of the location of sensor nodes. Let m number of obstacles are present in the monitoring area, where $m << \mathcal{N}$. The obstacles are detected with the

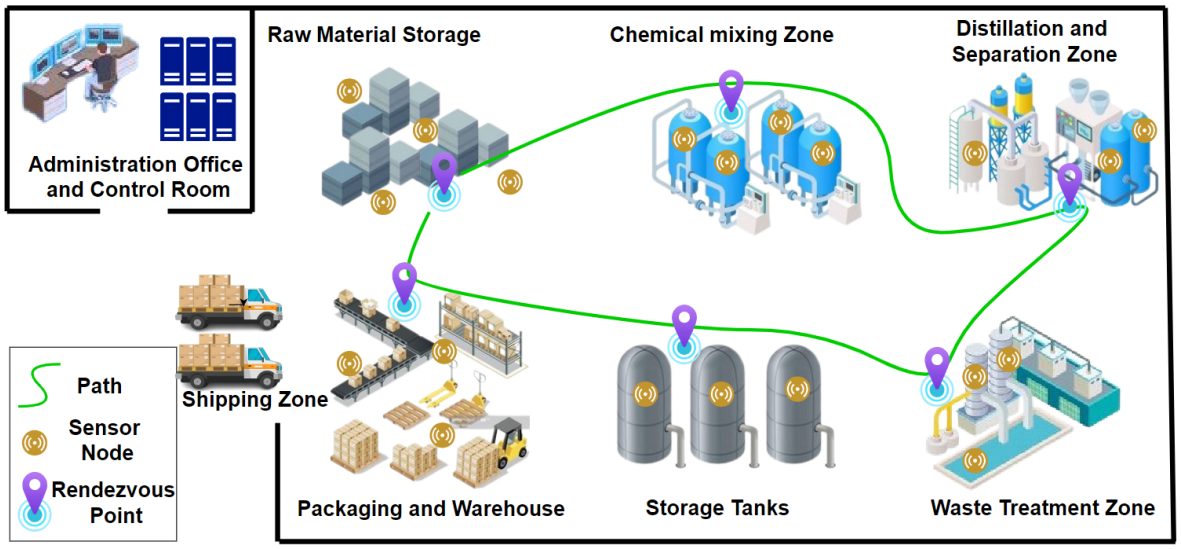


Figure 5.2: IIoT-based industrial safety monitoring in a smart chemical industry.

help of sensor nodes deployed around the boundary of obstacles [137]. The deployed sensor nodes are static, and an MS moves in the network at a constant speed. MS has adequate battery and storage space. The sensor nodes have limited energy and storage capacity. Two nodes communicate with each other only if the transmission distance is $\hat{\mathrm{D}}(s_i, s_j) \leq rc$. The communication links among nodes are symmetric. The terminologies used in this chapter are described in Table 5.1. A virtual grid is constructed over the deployment area. The grid cells are categorized into four different categories such as sensor cell, obstacle cell, empty cell, and RP cell. The cells which contain sensor nodes are called sensor cells. The cells that are covered by obstacles are called obstacle cells. The remaining cells that do not have either sensors or obstacles are called empty cells. The cells that are selected as RP are called RP cells. The empty cells are used for designing an obstacle-avoiding path for MS. This path is followed by MS to visit RPs for data collection from sensor nodes periodically.

Fig. 5.2 shows a chemical industry that includes a raw material zone, chemical mixing zone, distillation and separation zone, storage tanks, water treatment zone, administration office, control room, packaging and warehousing, and shipping area. In a smart chemical industry, multiple sensors, such as temperature, humidity, gas, vibra-

tion, and pressure sensors, are deployed for industrial safety and machine monitoring. These sensors continuously collect data from industrial machinery. An MS gathers data from the sensor nodes and delivers it to the BS. BS analyzes this data to assess machinery conditions. It helps the implementation of preventive maintenance programs and early detection of any hazardous events that occur within the industry. It reduces the likelihood of unexpected breakdowns, costly repairs, and unscheduled downtime. It also prevents major accidents which occur due to machine failures.

5.2.1 Energy model

In this paper, the energy model used to calculate the energy expenditure of the sensor nodes is similar to [67]. Each sensor node uses $\mathcal{E}_{tx}(l,d)$ amount of energy for transmitting l bit of data over the distance d. It also uses $\mathcal{E}_{rx}(l)$ amount of energy for receiving l bit data packet. The electronic circuit of a sensor node uses $\mathcal{E}_{elec}$ amount of energy. The $\mathcal{E}_{tx}(l,d)$ and $\mathcal{E}_{rx}(l)$ are calculated by the following equations.

$$\mathcal{E}_{(tx)}(l,d) = \begin{cases} l \times \mathcal{E}_{elec} + l \times e_{fs} \times d^2, & d < d_o \\ l \times \mathcal{E}_{elec} + l \times e_{mp} \times d^4, & d \geq d_o \end{cases} \tag{5.1}$$

$$\mathcal{E}_{rx}(l) = \mathcal{E}_{\text{elec}} \times l. \tag{5.2}$$

where e_{fs} is amplifier energy for the free space and e_{mp} is the energy for multi-path fading channel. The d_o is the threshold distance which is calculated as $\sqrt{\frac{e_{fs}}{e_{mp}}}$. The total energy consumed by a node in transmitting and receiving data is as follows.

$$\mathcal{E}_c(s_i) = pkt_{ij} \times \mathcal{E}_{tx}(l,d) + pkt_{ik} \times \mathcal{E}_{rx}(l). \tag{5.3}$$

where pkt_{ij} is the number of packets transmitted and pkt_{ik} is the number of packets received by s_i .

5.3 Problem Statement

This section defines the problem of designing an optimal obstacle-aware energy-efficient data collection path planning for MS in WSNs. The main objective of this work is to select an optimal set of RPs ($\mathbb{R}$) and design an optimal energy-efficient obstacle-aware smooth data collection path for MS. It must meet the following conditions.

1. To minimize the energy expended by sensor nodes in transmitting their data to MS.

2. To minimize the data collection delay.

The mathematical representation of the above objective is as follows:

$$\text{Minimize } Obj = \{\mathbb{L}, \mathcal{E}_c(s_i)\} \tag{5.4}$$

where

$$\mathbb{L} = \sum_{i=1}^{p}\sum_{j=1}^{p} \tau_{i,j}\hat{L}\left(\mathcal{R}_i, \mathcal{R}_j\right) + \hat{L}\left(\mathcal{R}_0, \mathcal{R}_1\right) + \hat{L}\left(\mathcal{R}_p, \mathcal{R}_0\right) \tag{5.5}$$

such that

$$\prod_{\mathcal{R}_i \in R} \mathcal{V}(\mathcal{R}_i) = 1. \tag{5.6}$$

$$\text{where } \mathcal{V}(i) = \begin{cases} 1, \text{ if } \mathcal{R}_i \text{ comes only once in MS path} \\ 0, \text{ otherwise} \end{cases}$$

$$\sum_{j=1}^{p} \mathcal{A}(i,j) = 1, \text{for each } i = 1, 2, 3...\mathcal{N}. \tag{5.7}$$

$$\text{where } \mathcal{A}(i,j) = \begin{cases} 1 & \text{if } \mathit{allot}(s_i) = \mathcal{R}_j \\ 0 & \text{otherwise.} \end{cases}$$

where $\tau_{i,j}$ is 1 if $\mathcal{R}_i$ comes immediately before $\mathcal{R}_j$ in the MS Path, otherwise 0.

$\hat{L}(\mathcal{R}_i, \mathcal{R}_j)$ is the length of the path between $\mathcal{R}_i$ and $\mathcal{R}_j$. The constraint (5.6) ensures that each RP appears only once in the MS path. Furthermore, it also ensures that all RPs are visited by MS in the data collection tour. The constraint (5.7) ensures that each sensor node is assigned to only one RP.

5.4 Proposed Work

The proposed work is divided into two phases. The first phase is the setup phase, and the second is the MS-based data routing phase. In the setup phase, the MRFO algorithm is used to identify the optimal positions of RPs and create optimal clusters in the WSNs. Furthermore, the EBS-A* algorithm is used to identify an obstacle-avoiding optimal smooth path for MS. In the MS-based data routing phase, the MS follows the optimal obstacle avoidance path and collects data from the deployed sensor nodes. The following sections provide a detailed description of the proposed work.

5.4.1 MRFO based optimal number of RP selection and cluster formation phase

The RPs selection and clustering are performed by using the MRFO algorithm. The first objective of the proposed work is to select optimal positions of RPs within the network. The proposed approach selects RPs in such a way that minimizes the Maximum Average Transmission Distance ($\mathcal{MAT}_{\mathcal{D}ist}$) among RPs. Sensor nodes lose a major amount of their energy for data aggregation and forwarding. Therefore, the proposed approach minimizes the Maximum Average Hop Count ($\mathcal{MAHC}$) of RPs to reduce the hops required for data transmission. It significantly reduces the energy loss of the sensor nodes for data aggregation and forwarding. Sensor nodes that are not reachable from RP directly send their data to RP through a neighbour node. These nodes then act as child nodes of the forwarding node. One sensor node having multiple child nodes increases the load on that node. Therefore, this scheme minimizes the Maximum number

of Child Nodes ($\mathcal{MCN}$) of a sensor node to balance the load. Furthermore, proposed approach also minimized the Maximum Distance ($\mathcal{MDC}$) of RPs from the network center. It helps to reduce the length of the MS path and overall data collection delay. The number of nodes assigned to $\mathcal{R}_j$, $\mathcal{R}_j \in \mathbb{R}$ is $\mathcal{N}_{\mathcal{SN}}(\mathcal{R}_j)$. It is calculated as follows.

$$\mathcal{N}_{\mathcal{SN}}(\mathcal{R}_j) = \sum_{i=1}^{\mathcal{N}} (\mathit{allot}(s_i) = \mathcal{R}_j)). \tag{5.8}$$

Parameter $\mathcal{SumT}_{\mathcal{D}ist}(\mathcal{R}_j)$ is the summation of transmission distances of sensor nodes which are assigned to $\mathcal{R}_j$. It is calculated as follows.

$$\mathcal{SumT}_{\mathcal{D}ist}(\mathcal{R}_j) = \sum_{i=1}^{\mathcal{N}} \mathcal{T}_{\mathcal{D}ist}(s_i) \times (\mathit{allot}(s_i) = \mathcal{R}_j)). \tag{5.9}$$

The average transmission distance to reach $\mathcal{R}_j$ is

$$\mathcal{AT}_{\mathcal{D}ist}(\mathcal{R}_j) = \frac{\mathcal{SumT}_{\mathcal{D}ist}(\mathcal{R}_j)}{\mathcal{N}_{\mathcal{SN}}(\mathcal{R}_j)} \tag{5.10}$$

Maximum Average Transmission Distance $\mathcal{MAT}_{\mathcal{D}ist}$ among all RPs is calculated as follows.

$$\mathcal{MAT}_{\mathcal{D}ist} = max\{\mathcal{AT}_{\mathcal{D}ist}(\mathcal{R}_j)\}, 1 \leq j \leq p. \tag{5.11}$$

Parameter $\mathcal{N}_{\mathcal{HC}}(\mathcal{R}_j)$ is the summation of hops required by sensor nodes that are allocated to $\mathcal{R}_j$. It is calculated as follows.

$$\mathcal{N}_{\mathcal{HC}}(\mathcal{R}_j) = \sum_{i=1}^{\mathcal{N}} \hat{\mathrm{H}}(s_i) \times (\mathit{allot}(s_i) = \mathcal{R}_j)). \tag{5.12}$$

The average hop counts to reach $\mathcal{R}_j$ is

$$\mathcal{AHopC}(\mathcal{R}_j) = \frac{\mathcal{N}_{\mathcal{HC}}(\mathcal{R}_j)}{\mathcal{N}_{\mathcal{SN}}(\mathcal{R}_j)} \tag{5.13}$$

The maximum average hop counts $\mathcal{MAHC}$ among all RPs is calculated as follows.

$$\mathcal{MAHC} = max\{\mathcal{AHopC}(\mathcal{R}_j)\}, 1 \leq j \leq p. \tag{5.14}$$

The maximum number of child nodes of a sensor node is computed as follows.

$$\mathcal{MCN} = max\{\mathcal{N}_{\mathcal{CN}}(s_i)\}, 1 \leq i \leq \mathcal{N}. \tag{5.15}$$

The maximum distance of an RP from the network center is calculated as follows.

$$\mathcal{MDC} = max\{\hat{D}\left(\mathcal{R}_j, \mathcal{C}_{xy}\right)\}, 1 \leq j \leq p. \tag{5.16}$$

where $\mathcal{C}_{xy}$ is the network center. MRFO is an optimization algorithm based on the foraging behaviour of Manta Rays. The manta rays use three foraging techniques such as chain foraging, cyclone foraging, and somersault foraging. It consumes minimum resources during the execution, which is very essential for resource-constraint WSNs. In this algorithm, these strategies are used to find the optimal solution. Initially, some random values are generated for a set of agents. The fitness value of each agent is computed to find the best agent. The foraging techniques are used to update agents' positions. Chain foraging is defined as follows.

$$\mathcal{P}_i^{t+1} = \begin{cases} \mathcal{P}_i^t + r \cdot (\mathcal{P}_b^t - \mathcal{P}_i^t) + \alpha \cdot (\mathcal{P}_{\mathrm{b}}^t - \mathcal{P}_i^t) & i = 1 \\ \mathcal{P}_i^t + r \cdot \left(\mathcal{P}_{i-1}^t - \mathcal{P}_i^t\right) + \alpha \cdot (\mathcal{P}_{\mathrm{b}}^t - \mathcal{P}_i^t) & i = 2, .., \mathcal{M}. \end{cases} \tag{5.17}$$

$$\alpha = 2 \cdot r \cdot \sqrt{|\log(r)|}. \tag{5.18}$$

where $\mathcal{P}_i^t$ is the position of i^{th} agent at the iteration t, and r$\in$[0, 1] is a random number. Parameter $\mathcal{M}$ is the size of the population, $\mathcal{P}_b^t$ is the position of the best agent, and α is the weight coefficient. Cyclone foraging is the spiral-shaped motion towards the food

by the manta rays. The cyclone foraging is represented as follows.

$$\mathcal{P}_i^{t+1} = \begin{cases} \mathcal{P}_b^t + r \cdot (\mathcal{P}_b^t - \mathcal{P}_i^t) + \beta \cdot (\mathcal{P}_{\mathrm{b}}^t - \mathcal{P}_i^t) & i = 1 \\ \mathcal{P}_b^t + r \cdot \left(\mathcal{P}_{i-1}^t - \mathcal{P}_i^t\right) + \beta \cdot (\mathcal{P}_{\mathrm{b}}^t - \mathcal{P}_i^t) & i = 2, .., \mathcal{M}. \end{cases} \tag{5.19}$$

$$\beta = 2e^{r_1(\frac{T-t+1}{T})} \times \sin(2\pi r_1). \tag{5.20}$$

where β is a weight coefficient, variable r_1 is a random number between [0,1], and T is total number of iteration. Eq. 5.19 shows the exploitation ability of the MRFO algorithm. The exploration ability of MRFO is represented as follows.

$$\mathcal{P}_i^{t+1} = \begin{cases} \mathcal{P}_r^t + r \cdot (\mathcal{P}_r^t - \mathcal{P}_i^t) + \beta \cdot (\mathcal{P}_{\mathrm{r}}^t - \mathcal{P}_i^t) & i = 1 \\ \mathcal{P}_r^t + r \cdot \left(\mathcal{P}_{i-1}^t - \mathcal{P}_i^t\right) + \beta \cdot (\mathcal{P}_{\mathrm{r}}^t - \mathcal{P}_i^t) & i = 2, .., \mathcal{M}. \end{cases} \tag{5.21}$$

$$\mathcal{P}_{\mathrm{r}}^t = \mathcal{LB} + r \times (\mathcal{UB} - \mathcal{LB}). \tag{5.22}$$

where $\mathcal{UB}$ and $\mathcal{LB}$ are upper and lower bounds of the search space. The last stage of the MRFO algorithm is somersault foraging. The following equation represents the somersault foraging.

$$\mathcal{P}_i^{t+1} = \mathcal{P}_i^t + \mathcal{S} \times \left(r_2 \cdot \mathcal{P}_b^t - r_3 \cdot \mathcal{P}_i^t\right) i = 1, 2, \ldots, \mathcal{M}. \tag{5.23}$$

where $\mathcal{S}$ is a somersault factor. r_2 and r_3 are two random numbers between 0 and 1. These three foraging techniques are applied to a population to find an optimal solution. Fig. 5.3 shows the flow diagram of the proposed MRFO algorithm. Population initialization and fitness computation are performed as follows.

- *Fitness Function:* The fitness function is used to select optimal RPs that minimize $\mathcal{MAT}_{\mathcal{D}ist}$, $\mathcal{MAHC}$, $\mathcal{MCN}$, and $\mathcal{MDC}$. The fitness function for the proposed MRFO

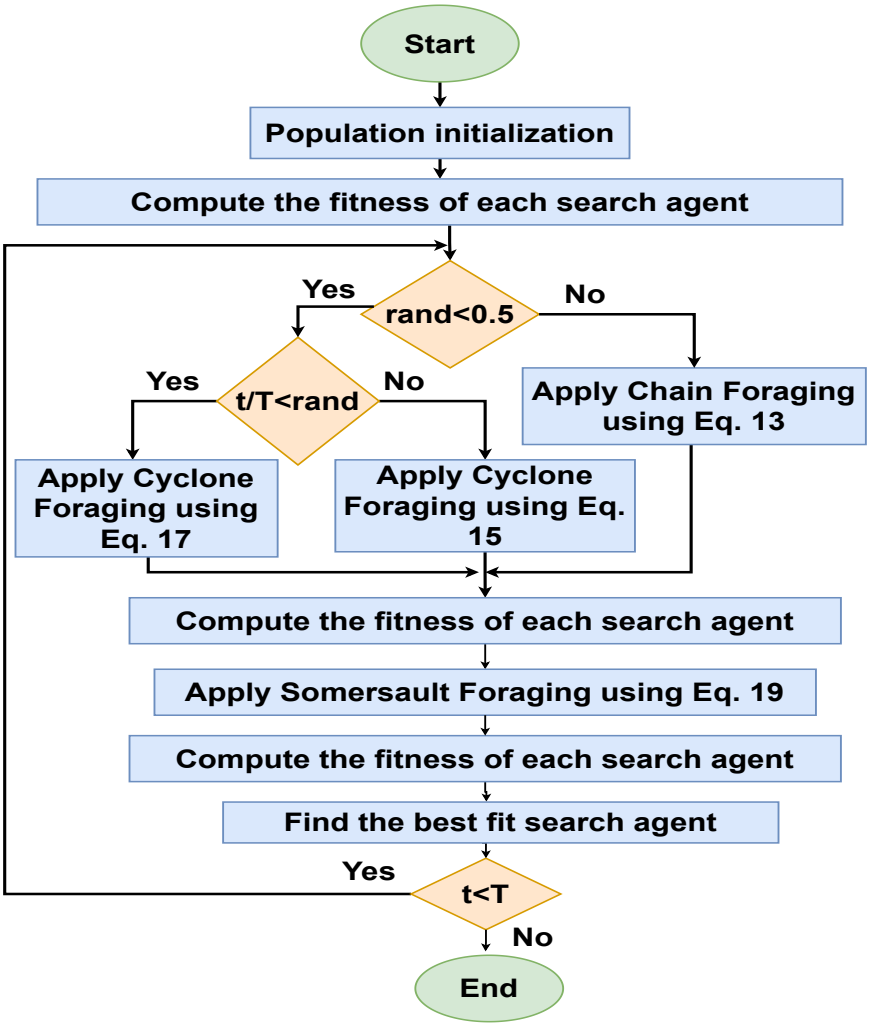


Figure 5.3: Flowchart of proposed MRFO algorithm.

algorithm is defined as follows.

$$Fitness = Minimize\{w_1 \mathcal{MAT}_{\mathcal{D}ist} + w_2 \mathcal{MAHC} + w_3 \mathcal{MCN} + w_4 \mathcal{MDC}\}. \tag{5.24}$$

Such that:

$$w_1 + w_2 + w_3 + w_4 = 1 \tag{5.25}$$

$$\hat{\mathrm{D}}(s_i, \mathcal{R}_j) \leq rc, \forall s_i \in \mathbb{S}, \forall \mathcal{R}_j \in \mathbb{R} \& (allot(s_i) = \mathcal{R}_j)) \tag{5.26}$$

$$\hat{\mathrm{D}}(s_i, s_k) \leq rc, \forall s_i, s_k \in \mathbb{S}, \forall \mathcal{R}_j \in \mathbb{R}, i \neq k \& (allot(s_k) = \mathcal{R}_j)). \tag{5.27}$$

Algorithm 4: Proposed MRFO-based Optimal Number of RP Selection Algorithm

Output: Position of RPs
Initialize the population $p_i \epsilon \mathcal{P}$, $\forall i \leq 1 \leq \mathcal{M}$
Calculate the fitness value of each search agent using Eq. **??**
Find the best fit solution $\mathcal{P}_{best}$
while $t < T$ **do**
  for *i=1 to* $\mathcal{M}$ **do**
    if *rand* $<$ *0.5* **then**
      Apply Cyclone foraging
      if $t/T <$ *rand* **then**
        Apply exploration technique using Eq. 5.21
      else
        Apply exploitation technique using Eq. 5.19
    else
      Apply Chain foraging using Eq. 5.17
  Calculate the fitness of each search agent by applying Eq. **??**
  Find the best fit solution $\mathcal{P}_{best}$
  for *i=1 to* $\mathcal{M}$ **do**
    Apply Somersault foraging using Eq. 5.23
    Calculate the fitness of each search agent by applying Eq. **??**
    Find the best fit solution $\mathcal{P}_{best}$
Return the best solution $\mathcal{P}_{best}$

where w_1, w_2, w_3, and w_4 are the weights assigned to the parameters. Eq. 5.26 confirms that any sensor node is assigned to an RP only if that RP is within the communication range of the sensor node. Eq. 5.27 confirms that a sensor node communicates to an RP through its closest sensor node if it does not have any RP within its communication range. Sensor nodes that are allocated to the same RP become members of the same cluster. The algorithm 4 shows the pseudo-code of the proposed MRFO-based optimal number of RP selection algorithm.

The algorithm 4 shows the pseudo-code of the proposed MRFO-based optimal number of RP selection algorithm. First, the population is initialised, and the fitness of each search agent is calculated. The search agent with the best fitness is selected as $\mathcal{P}_{best}$. Based on the value of *rand*, either cyclone foraging or chain forging is applied to

each search agent in the population. The fitness of each search agent is calculated, and $\mathcal{P}_{best}$ is selected. Next, somersault foraging is applied to each search agent, and $\mathcal{P}_{best}$ is selected based on fitness. These steps are iterative executed till maximum iteration and $\mathcal{P}_{best}$ value is returned for selected RPs.

5.4.2 Obstacle aware optimal path planning mechanism

In this stage, the visiting order of RPs is first designed, and next, EBS-A* is applied to design an obstacle-avoiding smooth path. The visiting order refers to the order in which MS visits RPs with minimum delay and collects data from deployed SNs. Genetic Algorithm (GA) is used to find the optimal RP visiting order with minimum cost. The GA maintains a population of diversified solutions. It helps in exploring various regions of search space and avoids getting trapped in local optima. GA designs an optimal visiting order of RPs with minimum tour length. To avoid the obstacles that are present on the path between two consecutive RPs, the EBS-A* algorithm is applied. The EBS-A* algorithm also uses minimum resources during execution, which is very helpful in the real-life implementation in resource-constraint WSNs. The EBS-A* algorithm designs a smooth obstacle-avoiding optimal path for MS. A* is a highly efficient heuristic algorithm that finds an optimal path with minimum cost. The algorithm selects the lowest f value grid cell at every step. The f is calculated as follows.

$$f(i) = g(i) + h(i). \tag{5.28}$$

where *g(i)* is the cost of the path which is formed between the source cell S_g to the current grid cell G_i. *h(i)* is the estimated cost of the path which is formed between the current cell G_i to the destination grid cell D_g.

EBS-A* algorithm is an improved A* algorithm. It adds smoothing, bidirectional search, and expansion distance to the path-designing process. The algorithm keeps a buffer space called expansion distance between the designed path and obstacles to avoid

collision between MS and obstacles. The length of expansion distance depends upon the size and speed of MS. Let MS be equivalent to a cylinder with radius rd in a robot operating system. The current speed of MS is $\mathcal{SP}_i$ and $\mathcal{SP}_{th}$ is the threshold speed. The maximum speed of MS is $\mathcal{SP}_{max}$. The expansion distance $\mathcal{E}(\mathcal{SP}_i)$ is computed as follows.

$$\mathcal{E}\left(\mathcal{SP}_i\right) = \begin{cases} rd & \mathcal{SP}_i \leq \mathcal{SP}_{th} \\ \frac{\mathcal{SP}_i}{\mathcal{SP}_{th}} rd & \mathcal{SP}_{th} < \mathcal{SP}_i \leq \mathcal{SP}_{max}. \end{cases} \tag{5.29}$$

The EBS-A* algorithm performs a bidirectional search. It concurrently searches from both directions, source node to destination node and destination node to source node. The search process ends when the forward and backwards search nodes become adjacent nodes. The forward and backward path fragments are joined together to form a collision-free path. Furthermore, the EBS-A* algorithm uses a Bézier curve to smooth the path. A Bézier curve is a parametric curve that comprises Bernstein polynomials. It is expressed as follows.

$$\mathcal{P}(t) = \sum_{i=1}^{q} p_i \mathcal{B}_{i,q}(t),\ t \in [0,1]. \tag{5.30}$$

where p_i represents control points, $\mathcal{B}_{i,q}(t)$ represents Bernstein polynomials, and q is degree of polynomials. $\mathcal{B}_{i,q}(t)$ is represented as follows.

$$B_{i,q}(t) = \frac{q!}{i!(q-i)!} t^i (1-t)^{q-i}, i = 0, 1, \ldots, q. \tag{5.31}$$

Bézier curve is used to improve path practicality and enhance the speed of MS by minimizing sharp turns. It is achieved by breaking down sharp 90° turns into smaller angles, thereby enhancing path smoothness. Fig. 5.4 shows the process of smoothing sharp turns in a path. Algorithm 17 depicts the pseudo-code of the proposed obstacle avoiding optimal path planning algorithm for MS. First, empty lists $Path_S$ and $Path_D$

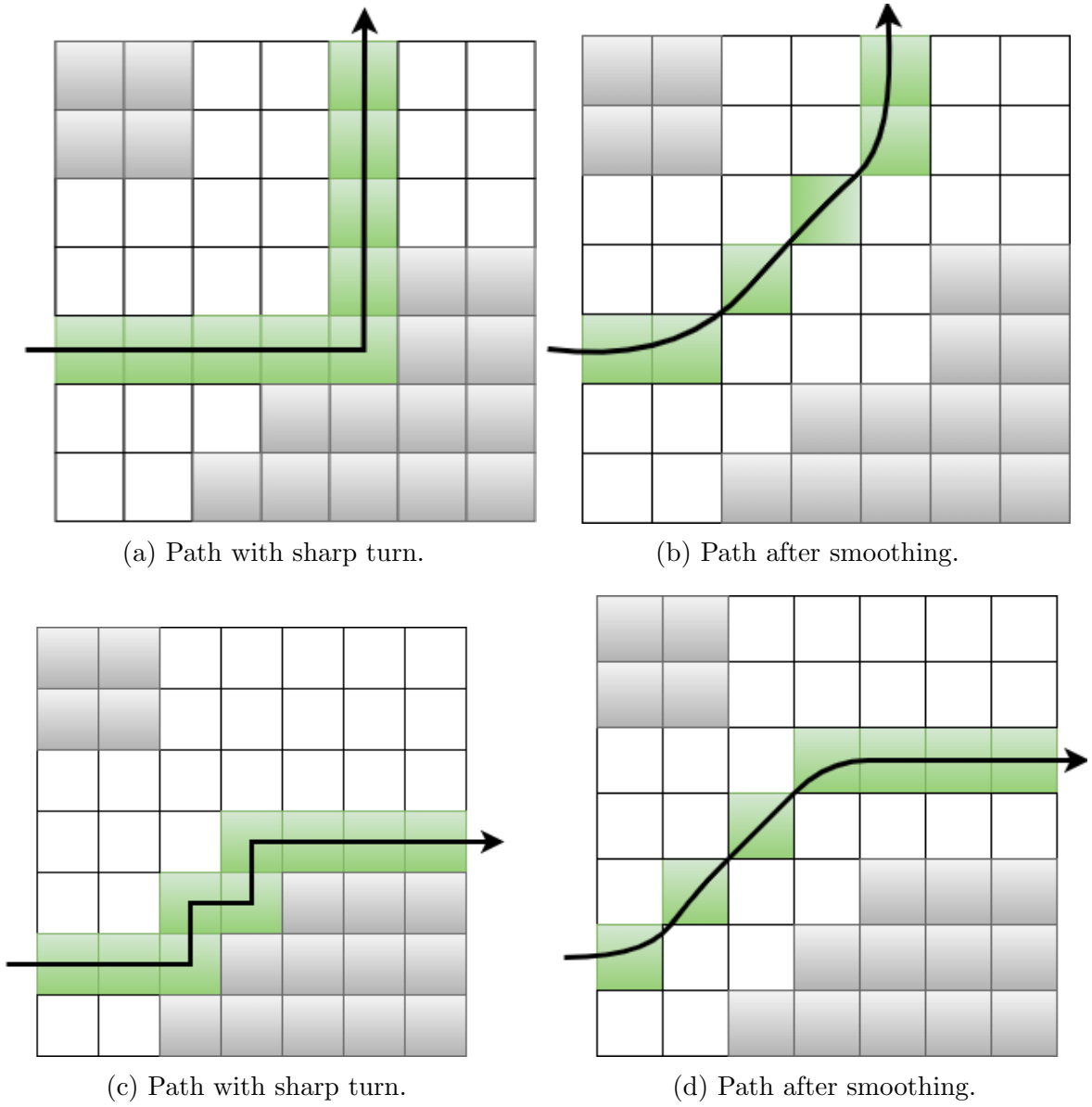
Algorithm 5: Obstacle Avoiding Optimal Path Planning Algorithm

Data: S_g, D_g
Result: $PATH$
Initialize $Path_S$, $Path_D$
Add expansion distance
Create forward open list $OPEN_S$ and forward close list $CLOSE_S$
Create backward open list $OPEN_D$ and backward close list $CLOSE_D$
Add S_g on the $OPEN_S$ and D_g on the $OPEN_D$
while $C_s \neq D_g \| C_d \neq S_g \| OPEN_S \neq \emptyset \| OPEN_D \neq \emptyset \| NG_i(C_s) \cap NG_i(C_d) == \emptyset$ **do**
    Set C_s = element with lowest f in $OPEN_S$
    Set C_d = element with lowest f in $OPEN_D$
    Remove C_s from $OPEN_S$ and C_d from $OPEN_D$
    Add C_s to $Path_S$ and add C_d to $Path_D$
    search(C_s, $OPEN_S$, $CLOSE_S$)
    Add C_s to $CLOSE_S$
    search(C_d, $OPEN_D$, $CLOSE_D$)
    Add C_d to $CLOSE_D$
$PATH = Path_S + reverse(Path_D)$
Smoothing the $PATH$
Return $PATH$

are created. Next, $OPEN$ and $CLOSE$ lists are created for both search directions. The source cell is added to $OPEN_S$, and the destination cell is added to $OPEN_D$. Cell with lowest f is searched and added to $Path$ in both search directions. The above steps are repeated until either $OPEN_S$ or $OPEN_D$ become empty, or either C_s finds D_g or C_d finds S_g. The algorithm is terminated if both C_s and C_d have a common neighbour cell. $Path_S$ and $Path_D$ are added together to design the obstacle-avoiding optimal path $PATH$ between two adjacent RPs. The algorithm is applied iteratively and selects an optimal obstacle-avoiding path for the MS with minimum cost.

Lemma 1: The time complexity of the proposed MS-based obstacle-aware data routing scheme is $O(T\mathcal{M}\mathcal{N} + I_{max}\mathcal{N}z + plw)$.

Proof: The complexity of the proposed MRFO-based optimal number of RP selection algorithm is $O(T \times \mathcal{M} \times \mathcal{N})$, where T is maximum iterations, $\mathcal{M}$ is population size, and $\mathcal{N}$ is number of sensor nodes. The worst case time complexity is $O(T\mathcal{M}\mathcal{N})$. Obstacle-avoiding optimal path design algorithm has two steps. The first step is finding an



(a) Path with sharp turn.

(b) Path after smoothing.

(c) Path with sharp turn.

(d) Path after smoothing.

Figure 5.4: Smoothing the path.

optimal visiting order using the genetic algorithm. The time complexity of the genetic algorithm is $O(I_{max} \times p \times z)$, where I_{max} is the maximum number of iterations, p is the size of each individual, i.e., the number of RPs and z is the size of the population. In the worst case scenario, $p = \mathcal{N}$. The worst-case complexity of finding the RP visiting order is $O(I_{max}\mathcal{N}z)$. The second step is finding an optimal path between RPs. The complexity of the EBS-A* algorithm is $O(l \times w)$, where l and w are the numbers of

rows and columns of the grid. EBS-A* algorithm is executed p times. Therefore, the total time complexity of the proposed scheme is $O(T\mathcal{MN} + I_{max}\mathcal{N}z + plw)$.

5.4.3 MS Safety Assessment

This section evaluates the performance of path planning approaches using a collision avoidance function, which is used to calculate the probability of MS collision with obstacles. MS safety assessment indicates the reachability of MS to all deployed sensor nodes with minimum delay during the data-gathering process. If the number of collisions increases, then MS is unable to reach all sensor nodes during data gathering. Furthermore, it also increases data collection delay. Therefore, collision avoidance is essential during MS-based data collection, primarily for industrial safety and security applications where QoS is critical. The distance function for MS safety assessment is calculated as follows.

$$d(a) = \min_{x,x'} \{\|x - x'\| \mid x \in A, x' \in O\} \tag{5.32}$$

Here, x represents a point in MS, and x' represents a point within an obstacle. $d(a)$ represents the minimum distance between an obstacle and MS at a random location a within the MS path. MS path is a set of points which is denoted by A, and obstacle is a set of points which is denoted by O. d is the shortest distance from a to the obstacle. The start and end points of the path are denoted by a_0 and a_T. The collision avoidance function at time t, where is $t \in [0, T]$, is computed as follows:

$$f(l) = \begin{cases} \frac{1}{d(a(t))} - \frac{1}{d_0}, & \text{for } d(a(t)) < d_0 \\ 0, & \text{for } d(a(t)) \geq d_0 \end{cases} \tag{5.33}$$

To assess the safety of the designed path, different norms are applied to the collision avoidance function. The L^p-norm is a mathematical function that calculates the dis-

tance or size of a vector in a certain space. It is computed as follows.

$$\|f\|_{(L^p)} \approx \left(\sum_{t=0}^{T} |f(t)|^p \cdot \right)^{\frac{1}{p}} \tag{5.34}$$

where p is a positive real number. The distance to an obstacle is measured by the L^p norm, which evaluates the distance between the MS and the obstacle. Each norm indicates the distance to obstacles, with higher values indicating closer proximity and high collision risk.

5.4.4 MS-based optimal data routing mechanism

In this phase, MS collects data from sensor nodes through the obstacle-avoiding optimal path. After reaching the RP, MS broadcasts an *announcement (ANN)* message to the cluster members in the cluster. *ANN* message includes the ID of the RP R_{id}. If any node receives *ANN* messages, it checks its R_{id}. If R_{id} matches, it indicates the node belongs to that RP and sends *acknowledgement (ACK)* messages to the MS. *ACK* includes node ID and buffer status of the node. Furthermore, MS assigns a time slot to each sensor node for data transmission based on received *ACK* messages. The sensor nodes send their data to MS according to the allotted time. If any sensor node does not receive an *ANN* message or MS does not receive *ACK* message due to transmission fault, then MS sends a final alert *(ALRT)* message within the cluster. The nodes that receive the final *ALRT* message check their current buffer condition. If the current buffer is empty, this sensor node has already sent its data to MS. Otherwise, the sensor node sends an *ACK* to MS. MS again allocates a time slot for data collection to the requesting node. MS moves to the next RP after the data collection. MS visits each RP and offloads data to the BS at the end of each data-gathering tour.

***Lemma 2*:** The Message complexity of the MS-based data routing mechanism is

$O(n)$.

***Proof*:** In the data routing phase, the MS broadcasts n number of ANN messages within the network. Similarly, n sensor nodes send ACK to the MS, and MS sends allocated time slots to sensor nodes. n sensor nodes also send their data to MS. Thus, the total message complexity of the proposed MS-based data routing mechanism is $O(n+n+n+n) \approx O(n)$. Low message complexity reduces network overhead and improves overall network performance.

5.5 Performance Analysis

This section evaluates the performance of the proposed MS-based obstacle-aware energy-efficient data routing scheme for IIoT-enabled WSNs. The simulations have been conducted on the NS3 simulator. Table 5.2 shows the parameter values which are used in the simulation. Fig. 5.5 shows the obstacle-aware optimal path designed by the proposed approach in various distinct scenarios that are simulated in indoor and outdoor environments. In the figure, the white region represents space where MS can move. The black blocks represent obstacles such as walls, containers, and other industrial equipment. The red blocks represent RPs, and the green blocks represent the designed path. The proposed approach designs an obstacle-avoiding optimal path for MS that easily avoids obstacles without any collision. The proposed scheme is compared with the state-of-the-art OCRS-MD approach [81], DGOB [67], and RRTMT algorithm [39] in terms of network lifetime, stability period, residual energy, data collection delay, and MS safety assessment.

The stability period is the duration between the network's starting work and the death of the first node in the network. Fig. 5.6a shows the network stability period in different node densities. Fig. 5.6b shows the stability period with different numbers of obstacles in different shapes. The simulation results depict that the proposed approach increases the network stability up to 39% than DGOB, 35% than OCRS-MD, and 30%

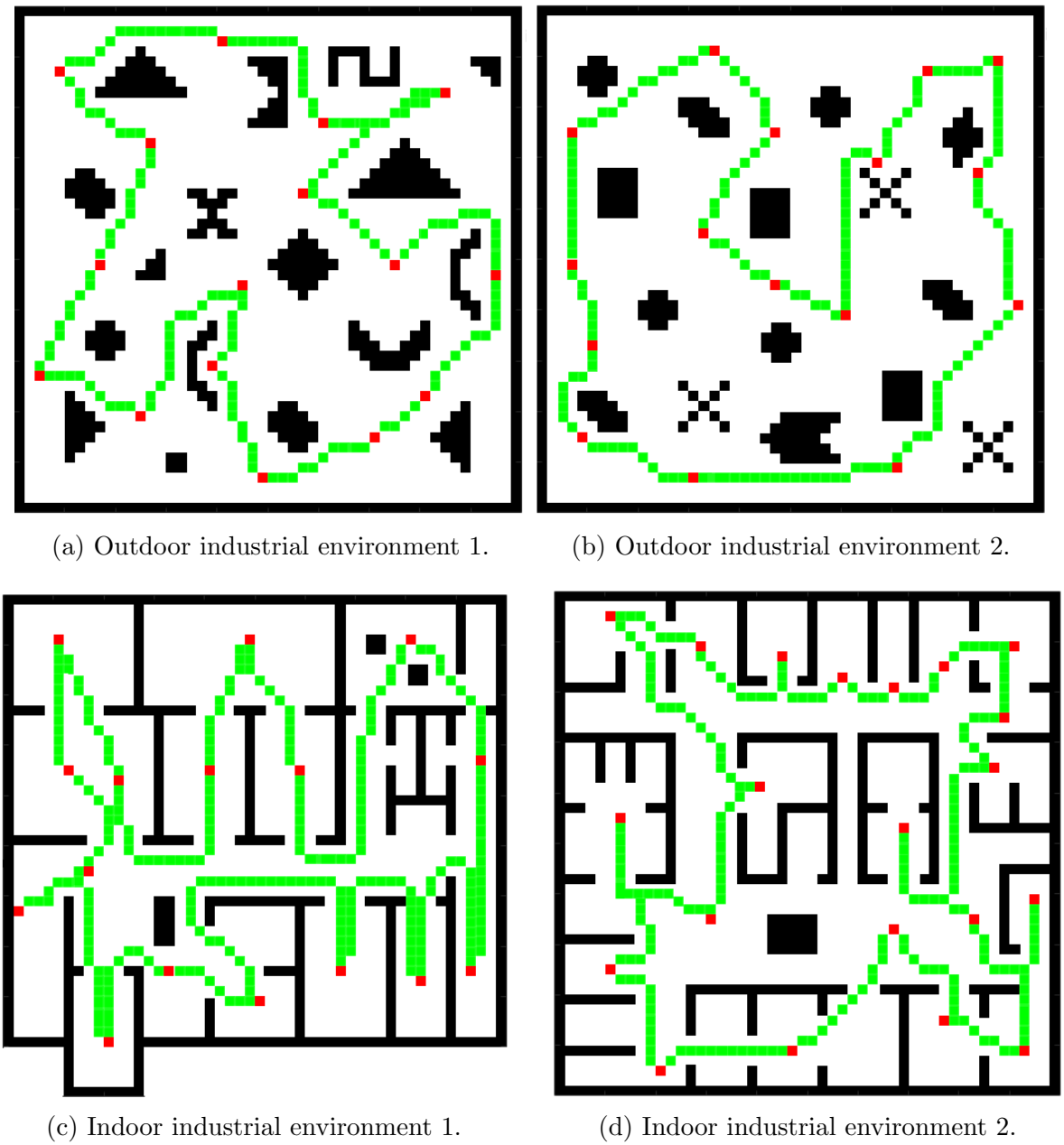
Table 5.2: Simulation parameters

Parameters	***Values***
WSN deployment area	200 X 200 m^2
Number of sensor nodes	100-400
Initial energy of each sensor node	1 J
$\mathcal{E}_{\text{elec}}$	50 nJ/bit
ε_{fs}	10 pJ/bit
ε_{mp}	0.0013 pJ/bit
MRFO Population size	40
Maximum Number of iterations MRFO	500
Speed of MS	2 m/s

than the RRTMT algorithm. This is because the proposed scheme identifies optimal RPs that reduce the distance and hop counts between sensor nodes and RPs. The proposed scheme also minimizes the load of sensor nodes by minimizing the child nodes, which considerably reduces sensor nodes' energy expenditure. Hence, the proposed scheme achieves better network stability as compared to existing methods.

The network lifetime of a WSN is the time from when the network starts operating till all nodes are dead. Fig. 5.7a shows the network lifetime in different node densities. Fig. 5.7b shows the network lifetime with a different number of obstacles in different shapes. The simulation results illustrate that the proposed approach extends network lifetime by 38% than DGOB, 34% than OCRS-MD, and 28% than RRTMT algorithms. The proposed approach performs better than the existing approaches in terms of network lifetimes due to optimal RP selection and optimal path design for MS-based data gathering. Optimal RP selection and clustering significantly reduce the energy consumption of sensor nodes. It increases the lifetime of sensor nodes as well as the overall network lifetime.

Fig. 5.8a shows the total residual energy in different node densities. Fig. 5.8b depicts the total residual energy with different numbers of obstacles in different shapes. These figures illustrate that the proposed approach enhances the residual energy at sensor nodes by 41% than DGOB, 36% than OCRS-MD, and 30% than the RRTMT algorithm. This is because the proposed scheme selects optimal RP positions that minimize the data transmission energy loss. The proposed scheme minimizes hop counts



(a) Outdoor industrial environment 1.

(b) Outdoor industrial environment 2.

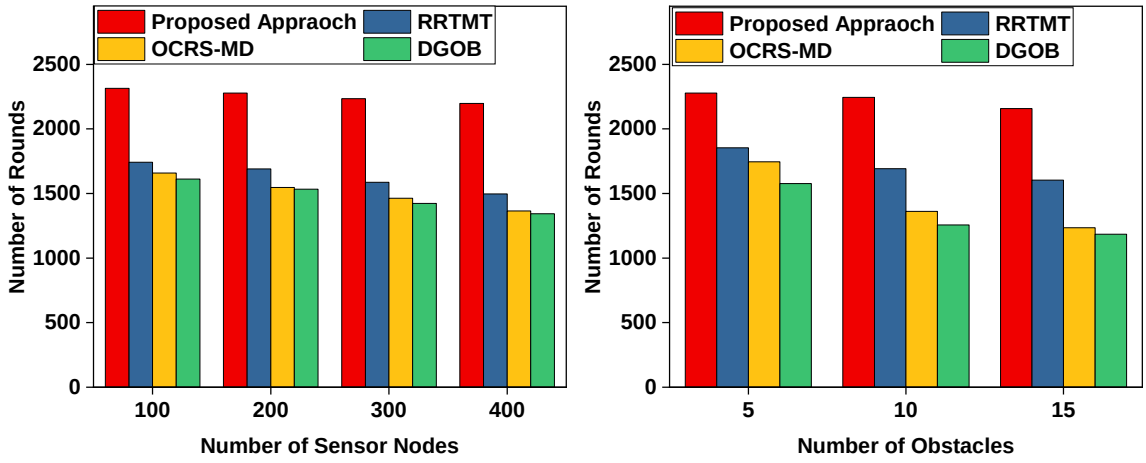
(c) Indoor industrial environment 1.

(d) Indoor industrial environment 2.

Figure 5.5: Obstacle aware path design for various environments.

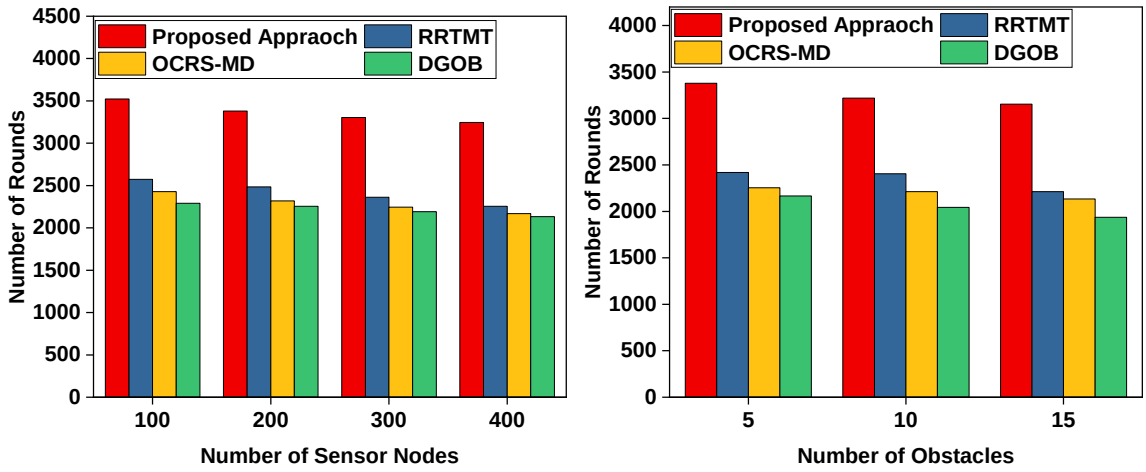
between RPs and sensor nodes. It reduces the energy used in data aggregation and forwarding at sensor nodes. Furthermore, the proposed approach designs an optimal obstacle avoidance data collection path for MS that significantly minimizes the energy consumption of sensor nodes.

Fig. 5.9a depicts the data collection delay in different node densities. Fig. 5.9b



(a) Varying number of sensor nodes. (b) Varying number of obstacles.

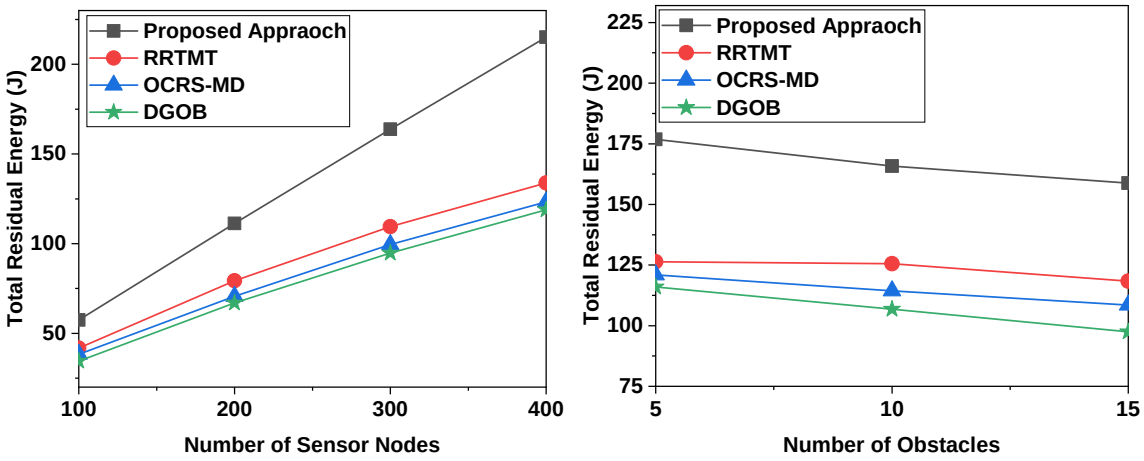
Figure 5.6: Stability period.



(a) Varying number of sensor nodes. (b) Varying number of obstacles.

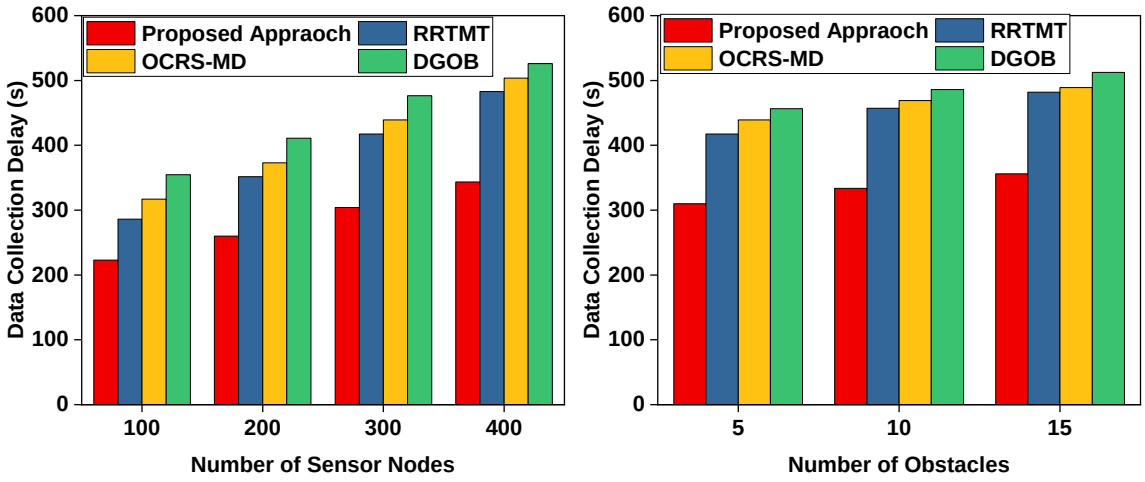
Figure 5.7: Network lifetime.

shows the data collection delay in different numbers of obstacles. The simulation results show that the proposed approach lessens the delay by 35% than DGOB, 30% than OCRS-MD, and 26% than RRTMT. The proposed approach selects RPs close to the network center. It minimizes the length of the MS path. The proposed approach also designs a smooth path using the EBS-A* algorithm that minimizes the MS tour length. The smooth path enables smooth MS movement, which decreases the data collection



(a) Varying number of sensor nodes. (b) Varying number of obstacles.

Figure 5.8: Residual energy.



(a) Varying number of sensor nodes. (b) Varying number of obstacles.

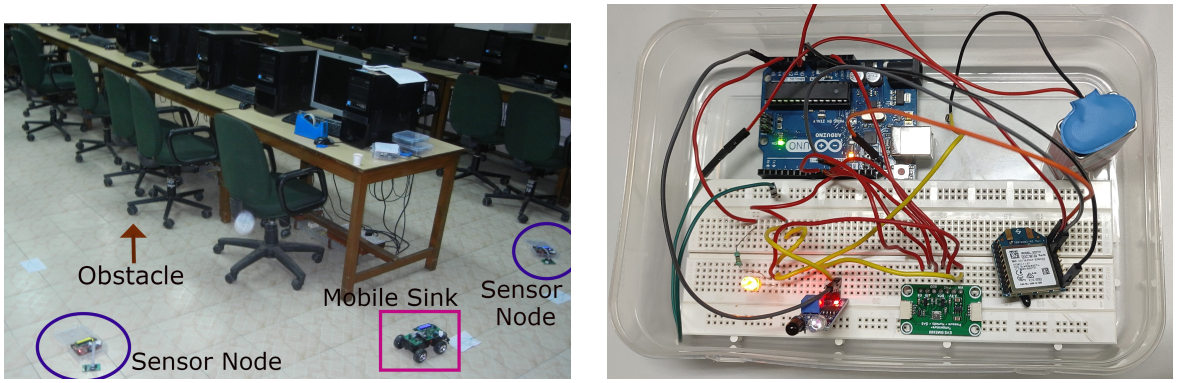
Figure 5.9: Data collection delay.

delay. The proposed approach maintains a safe distance from obstacles that prevent collisions and minimizes the data collection delay. It also indicates that the proposed scheme collects data from all the deployed sensor nodes due to the increased reachability of the MS.

Table 5.3 shows the performance of the proposed approach for MS safety assessment. L^1 norm shows the risk of collision depending on the average distance to obstacles. The

Table 5.3: MS safety assessment

	DGOB	*OCRS-MD*	*RRTMT*	*Proposed Approach*
L^1	62.71	59.49	55.39	29.69
L^2	6.33	6.15	5.98	4.2
L^∞	0.73	0.65	0.65	0.46



(a) Sensor nodes and MS.

(b) Sensor nodes used in the testbed.

Figure 5.10: Testbed.

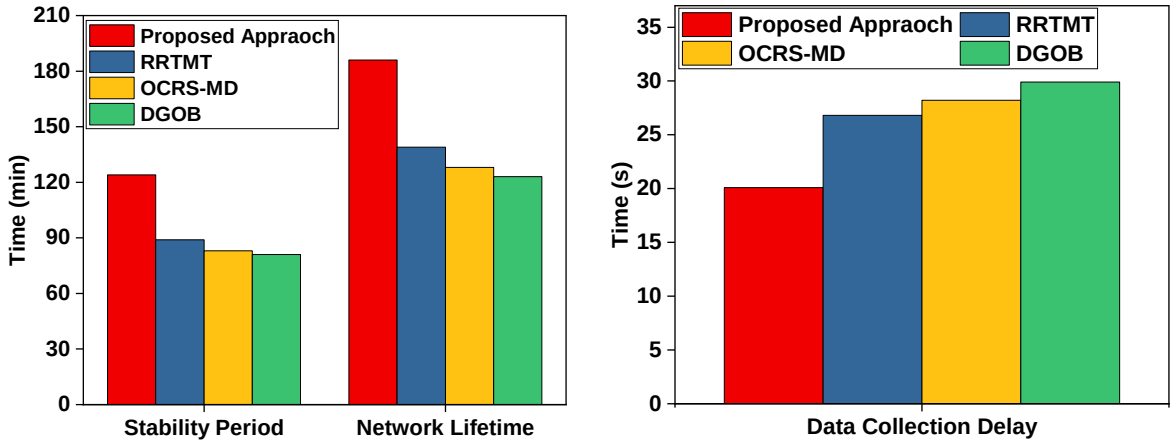
proposed approach shows the L^1 norm 52% less as compared to DGOB, 50% less as compared to OCRS-MD and 46% less as compared to RRTMT approaches. L^∞ norm shows the risk of collision depending on the shortest distance to an obstacle. The proposed approach shows the L^∞ norm 58% less as compared to DGOB and 41% less as compared to OCRS-MD and RRTMT approaches. The proposed approach uses the EBS-A* algorithm that designs a smooth path for MS movement. Furthermore, the proposed approach keeps a safety margin between the designed path and obstacles that minimizes the probability of collision. The minimum collision also increases the reachability of the MS. It also indicates that the proposed scheme is able to collect data from all of the deployed sensor nodes with minimum delay. It is very essential for industrial safety and security applications.

5.5.1 Testbed experiment

This work conducts testbed experimentation to evaluate the performance of the proposed scheme in the real environment.The experiment is conducted in an indoor envi-

Table 5.4: Average residual energy

Time (min)	*Proposed Approach*	*RRTMT*	*OCRS-MD*	DGOB
30	10.24	10.21	10.17	10.06
60	9.69	8.68	8.14	8.12
90	8.86	6.83	6.42	6.18
120	7.78	4.87	4.56	4.24
150	6.79	3.22	2.83	2.31



(a) Stability period and network lifetime.

(b) Data collection delay.

Figure 5.11: Testbed results.

ronment, where nine sensor nodes are deployed in a room of 9x6 $[m^2]$ dimension. Walls and room furniture act as obstacles for this testbed experiment. Fig. 5.10a shows the deployment of the testbed. In this testbed, sensor nodes are composed of Arduino UNO with an XBee module for data transmission. The sensor nodes are equipped with temperature, humidity, gas, pressure, and IR sensors that collect data from the environment. Fig. 5.10b shows the sensor node used in the testbed. A robotic vehicle is used as an MS, and a laptop is used as the BS. Table 5.4 shows the average residual voltage of sensor nodes at different time intervals. The result shows that the proposed approach lessens energy expenditure up to 40% than DGOB, 34% than OCRS-MD, and 28% than RRTMT approaches. The result shows that the proposed approach conserves more energy as compared to OCRS-MD, DGOB, and RRTMT approaches, similar to the simulation result. This is because of MRFO algorithm-based optimal clustering of sensor nodes. Furthermore, the proposed approach minimizes the transmission distance

and hop counts, which reduces the sensor nodes' energy expenditure. Fig. 5.11a shows the network lifetime and stability period. The proposed approach extends the stability period by 37% than DGOB, 33% than OCRS-MD, and 28% than RRTMT approaches similar to the simulation result. This is because the proposed scheme minimizes the load of a sensor node by minimizing the number of child nodes, which decreases the energy expenditure of sensor nodes. It helps in saving energy at sensor nodes and prolongs the stability period. The result indicates that the proposed approach extends the network lifetime up to 35% than DGOB, 32% than OCRS-MD, and 26% than RRTMT approaches. In the real environment, the proposed scheme depicts a higher network lifetime than the other three approaches, similar to the simulation results. This is because the proposed obstacle-avoiding data routing scheme reduces the energy expenditure of the sensor nodes and increases the network's lifetime. Fig 5.11b displays the data collection delay. The proposed approach reduces the delay by 33% than DGOB, 28% than OCRS-MD, and 25% than RRTMT approaches. This is because the proposed approach finds the minimum cost path using EBS-A* algorithm. The proposed approach also minimizes the jags in the path that create a shorter MS tour that minimizes the data collection delay.

5.6 Summary

This chapter proposes an MS-based optimal obstacle-aware energy-efficient data routing scheme for IIoT-enabled WSNs. The proposed scheme applies the MRFO algorithm to select an optimal number of RPs. It also forms optimal clusters within WSNs. Furthermore, the EBS-A* algorithm is used to design a smooth obstacle-avoiding optimal data-gathering path for MS. It designs the shortest data-gathering path for MS that keeps a safe distance between the designed path and obstacles. It reduces the risk of MS collision with obstacles during data collection. It also designs a smooth path that enables smooth MS movement in the network. Extensive simulation and testbed

experiments have been performed to analyze the performance of the proposed scheme. The proposed approach increases the network stability up to 39% than DGOB, 35% than OCRS-MD, and 30% than the RRTMT algorithm. Furthermore, the proposed approach extends network lifetime by 38% than DGOB, 34% than OCRS-MD, and 28% than RRTMT algorithms. The proposed approach also enhances the residual energy at sensor nodes by 41% than DGOB, 36% than OCRS-MD, and 30% than the RRTMT algorithm. In terms of data collection delay, the proposed approach lessens the delay by 35% than DGOB, 30% than OCRS-MD, and 26% than RRTMT. The simulation and testbed results illustrate that the proposed scheme performs better as compared to the existing state-of-the-art algorithms.

As per the literature, network cut is one of the major issues for degradation of WSNs performance. Therefore, an efficient network cut detection and recovery scheme can significantly improve overall network performance as well as the lifetime of the network. The next chapter presents a mobile data collector-based novel network cut detection algorithm and network recovery algorithm for IoT-enabled WSNs.