



An adaptive coverage aware data gathering scheme using KD-tree and ACO for WSNs with mobile sink

Zaher Al Aghbari¹ · Ahmed M. Khedr¹ · Banafsaj Khalifa¹ · Pravija P. V. Raj¹

Accepted: 24 February 2022 / Published online: 21 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

While several Mobile Sink (MS)-based data gathering methods have been proposed for Wireless Sensor Networks, most of them are less adaptive to changes in network topology, and the planned MS trajectory cannot be refined to accommodate node failures. Hence, a KD-Tree-based scheme (KDT) is proposed, which is an adaptive and robust algorithm that reduces network energy consumption and data gathering delay. In contrast to many existing algorithms, the number of Rendezvous Points (RPs) are assigned dynamically by prioritizing the nodes' coverage. Overlapping coverage of RPs is minimized, while guaranteeing 100% coverage of nodes. KDT is adaptive to network topology changes, and the planned MS trajectory can be refined to accommodate node failures. The shortest MS path is found using Ant Colony Optimization. Simulation results show that KDT requires approximately half the number of RPs and about 13% reduction in MS travel time compared to existing schemes.

Keywords Wireless sensor networks · KD-tree · Travel route planning · Trajectory design · Data gathering · Mobile sink · Ant colony optimization

1 Introduction

Wireless sensor actuator networks (WSANs) consist of several nodes that can sense aspects of the physical environment and actuators that can act upon the incoming sensing data [1, 2]. Typically, sensor nodes are small resource constrained battery powered devices that observe phenomena to acquire data, while the actuators are resource-rich entities that can take action to change aspects of their environment based on incoming data. This interaction forms a closed-loop system in the WSAN which allows for a wide range of applications through computation and control in

✉ Zaher Al Aghbari
zaher@sharjah.ac.ae

¹ Department of Computer Science, University of Sharjah, Sharjah 27272, United Arab Emirates

large scale networks [3–5]. WSAWs are emerging as a new generation of Wireless Sensor Networks (WSNs), and they face much of the same challenges. Sensor nodes are typically deployed to cover a predetermined geographical area, and after some network processing, forward the data to a fixed sink via multi-hop routing [6]. However, this approach can give rise to many issues. Sensor nodes have limited energy, and it is critical to keep energy consumption as low as possible. But multi-hop routing causes a significant increase in energy consumption to relay data from other nodes and forward them to the sink node. Data forwarding causes uneven energy distribution as the nodes closer to the sink must forward data from all over the network, which means that nodes closer to the sink use considerably more energy than those situated farther away. Depending on the routing method, these nodes may rapidly deplete their energy and die out, while other nodes survive for a longer time. If that happens, the network becomes disconnected and results in the sink hole problem [7, 8].

Nowadays, a common solution is to use an actuator in the form of a Mobile Sink (MS) which moves through the network and collects the data, thereby removing the need for excessive data forwarding [9, 10]. With this, sensors consume less energy because data forwarding is minimized, and energy distribution is more balanced because of the reduction of routing overhead. Even if part of the network becomes disconnected for any reason, the MS can still reach those sensor nodes and gather data from them. On the other hand, this solution also introduces a new set of design challenges. MS needs time to travel to sensors and gather data from them which is called the data gathering delay. It consumes more time to gather data than the traditional multi-hop routing as the velocity of a MS is slower than transmission time by several orders of magnitude. Therefore, it is important to have the shortest travel route possible in order to improve the network lifetime and to reduce the latency in data collection. Also, the route must be carefully designed to ensure that sensor nodes will be within communication range of the MS at some point on the tour so that they can upload their data in single hop communication, which will in turn contribute to maximizing the network lifetime. Even though researchers have studied this problem from various aspects, most of the algorithms are not adaptive to any changes in network topology, and the planned MS trajectory cannot be refined to accommodate node failures.

Motivated by this, an adaptive and robust algorithm called KD-Tree-Based Scheme (KDT) is proposed to reduce network energy consumption and minimize data gathering delay. The algorithm is adaptive to any changes in network topology, and the planned MS trajectory can be refined to accommodate node failures. In contrast to many existing algorithms, the number of Rendezvous Points (RPs) are not pre-decided; it is assigned dynamically depending on the number and location of deployed nodes. This allows to create shorter travel routes and minimize data gathering latency. Moreover, instead of polling points, RPs are set to unique geographical positions to enable the MS to access as many sensor nodes as possible in a single hop communication, eliminating the need for data forwarding and, as a result, allowing sensors' resources to be conserved. In this work, a KD-Tree search-based algorithm (KDT) to help with locating the best positions of RPs. First, the KD-tree is constructed based on node deployment in the

sensor field [11, 12]. With the help of the KD-tree, the positions of nearby sensor nodes are efficiently searched to allow the placement of RPs at best locations to cover the maximum number of sensor nodes. This allows the sensor nodes to require only single hop communication to upload their data to the MS, which also reduces the energy consumption across the whole network. To ensure that overlapping coverage of RPs is minimized and to prevent outliers on the edge of the sensor field, the RPs are initially located on the edge of the sensor field, and then moved inwards. When the position of the RPs is finalized, Ant Colony Optimization (ACO), which is effective in solving optimization problems, is utilized to find the shortest path for MS to gather data. This paper provides the following contributions.

- An adaptive and robust algorithm is proposed in which the number of RPs are not predetermined, but dynamically allocated as needed based on sensor deployment. This allows the location and number of RPs to easily adapt to changes in the network topology.
- The algorithm reduces the number of RPs when node failures are detected allowing it to create shorter travel routes and minimize data gathering latency.
- Extensive simulations show that KDT effectively reduces data gathering time while also minimizing communication energy consumption as compared to similar recent algorithms.

The rest of the paper is organized as follows. The existing data gathering schemes are reviewed in Sect. 2. Then, the problem statement is discussed in Sect. 3. Next, the steps of the proposed algorithm are detailed in Sect. 4. Section 5 discusses the simulation experiments and their results. Finally, Sect. 6 concludes this paper. Table 1 summarizes the notations used in this paper.

Table 1 Table of abbreviations

Notation	Definition
WSN	Wireless sensor network
ROI	Region of interest
RP	Rendezvous point
R_s	Sensing range
R_c	Communication range
SP	Starting point
SN	Sensor node
MS	Mobile sink
ACO	Ant colony optimization

2 Related work

In recent literature, there have been many proposed solutions on how to design trajectories for effective data gathering in a WSN environment with a mobile sink. Solutions include such techniques as game theory, evolutionary algorithms, approximation heuristics, and optimization techniques [9]. The authors of [13] introduce an algorithm called multi-hop weighted revenue which aims to minimize the latency of data collection in multi-hop networks with the help of a MS. By considering the volume of concurrent uploaded data, as well as the planned path length and number of neighbors, the algorithm creates a weighted metric which can be used to rank the deployed sensor nodes for their eligibility as polling points. After polling point selection, an approximate algorithm is run for the traveling salesman problem (TSP) to find the shortest moving tour for the MS. Meanwhile, in [14], the authors aim to minimize the data delivery latency in a WSN by employing a MS. First, the anchor points for the MS are placed on the edge of nodes' communication range to shorten the MS traveling route. Furthermore, sensor nodes upload their data to the MS with time division slots. The authors formulate the delay latency minimization problem as an integer programming problem and solve it using three heuristic algorithms.

While these heuristic approaches reduce data delivery delay, they fail to prioritize reduction of energy consumption. The proposed algorithm differs in that it guarantees single hop communication for all nodes, thus reducing overall network energy consumption. In [15], the authors present a game theory-based route selection technique. It is a distributed method in which the MS gathers data by selecting rendezvous points using game theory and then planning a trajectory using ant colony optimization. The presented algorithm aims to reduce energy consumption and data delivery delay. Meanwhile, the authors of [16] present three different path planning algorithms based on application requirements: Reduced Energy Path (REP), Reduced Delay Path (RDP), and Delay Bound Path (DBP). In [17], the authors propose a grid-based scheme in which a virtual grid is constructed. It is then used to map the geographical locations of the sensor nodes and the MS to virtual addresses. Cell heads are elected for each grid, and they are responsible for forwarding sensed events to the MS. The authors claim that network energy consumption is decreased by utilizing this hierarchical functionality with optimal grid sizes. Meanwhile, the authors of Wang et al. [18] propose a clustering method for the sensor nodes followed by a MS touring the elected cluster heads. Clustering is performed using an improved artificial bee colony (ABC) algorithm. Factors such as the cluster heads' energy, density, and locations are considered to solve the clustering problem. Then, ant colony optimization (ACO) is used to determine the tour between the cluster heads and base station. On the other hand, the authors of Park et al. [19] introduce an iterative clustering method which focuses on the problem of "hotspots" that form if sensors are deployed more densely in some areas compared to others. The algorithm considers sensor density and residual batteries of sensors to find a suitable number of clusters. Then, the clusters and cluster heads are further segmented into sectors which

are assigned to one MS each. After that, ACO is applied to each sector to find the shortest trajectory for each MS. Conversely, the authors of Wang et al. [20] perform data fusion using a neural network to improve WSN performance. To accomplish this, the sensor field is partitioned into virtual grids. Then, cluster heads (CHs) are chosen, and data fusion is carried out on the CHs by utilizing a pretrained neural network. A mobile collector is then dispatched to gather information from the CHs.

While clustering achieves many goals, it also introduces overhead for the nodes that need to collaborate and elect a cluster head. In this work, this overhead is eliminated by selecting RP locations without the need for clustering. These locations are selected such that sensor nodes can send data directly to the MS in a single hop. The authors of Han et al. [21] propose that the MS should gather data both when it is stopped and when it is in motion. They named these communication phases as Parking Communication and Moving Communication, respectively. In their paper, they aim to maximize the amount of data collected by the MS in these two communication phases. They formulate the problem as two different optimization models under varying constraints. Although the presented algorithm reduces data delivery delay, data gathering is only maximized under the limited constraints presented. It does not guarantee that data will be collected from all the sensor nodes. The authors of Gao et al. [22] discuss a travel route planning schema with a mobile collector (TRPMC) in which they present the path planning problem as a coverage problem. The Sojourn Points (SPs) of the mobile collector (MC) are represented by their location and the communication range of the MC. The coverage problem is then defined so that the SPs must contain as many sensors as possible. Sensor nodes within communication range of a SP can relay their data to the MC in a single hop. To optimize the location of the SPs, Particle Swarm Optimization (PSO) is used. Then ACO is utilized to determine the shortest loop through the SPs. The route is planned only once at network startup and remains the same throughout the network's lifetime. In [23], the authors propose an ACO-based algorithm (ACO-MSPD) for selecting the RPs and planning the MS tour. The work utilized directed spanning tree to find the forwarding load of each nodes. They adopt re-selection of rendezvous points to balance the energy expended by the nodes and aim to maximize the network lifetime and minimize the data collection delay. Donta et al. [24] put forward an extended version, called extended ACO-MSPD (eACO-MSPD). They introduced the use of virtual RPs to reduce the data transmission communication between the nodes and RPs. The results indicate that eACO-MSPD performed better in terms of energy consumption and lifetime than the existing approaches. In [25], the authors consider the problem of data gathering in a WSN whose MS has limited mobility. To optimize network performance in such a case, the authors prioritize prolonging network lifetime and maintaining an acceptable performance with respect to data gathering. In their work, the network region is represented by a graph in which sensor nodes within the same grid are considered as a single unit. Then, each grid is assigned its own sub-sink according to a heuristic algorithm. Their proposed data gathering scheme considers the energy efficiency of hierarchical sub-sinks as well as the communication range of the nodes. The work in [26] utilize multi-objective PSO to reduce the MS tour length and balance the load of RPs. Potential stop points are

selected within the overlapping communication range of nodes, instead of nodes' locations. Then, a path encoding method is designed to generate the MS path that contains an unfixed number of stop points. The results indicate that it performs better in terms of expended energy, data delivery latency, and lifetime of WSN. In [27], the initial selection of RPs is based on distance between two adjacent RPs and data packets count. Convex polygon is utilized for trajectory planning of MS. In [28], the authors perform the clustering using a kd-tree-based algorithm to maintain equal energy utilization among the nodes. The authors of Dash et al. [29] introduce an algorithm for designing a data gathering tour that aims to minimize data gathering delay. It works by considering deployed sensors' locations and communication overlaps as well as data availability. The authors propose that higher data collection can be achieved in less time by optimizing the speed and data receiving schedule of the MS. Specifically, the communication disks of the sensors divide the sensing field into faces. The communication overlap between sensors forms intersection sets that are considered potential faces to be visited by the MS. The set of faces with the highest communication coverage are selected as visiting faces. The visiting points are calculated as the center of the boundary points of each visiting face. If it is an isolated sensor, then the position of the sensor is the visiting point. The MS follows a path through all the visiting points.

As seen from the previous researches, a lower number of RPs is preferable because fewer stop points means a shorter travel route and lower data gathering delay. However, this can cause several sensor nodes to be left uncovered by a RP, and the rest of the sensors must resort to multi-hop communication to get their data uploaded to the MS, which, in turn causes network energy consumption to increase. Also, the total trip time of the MS is affected by the traveling distance and number of RPs. Therefore, a robust and adaptable algorithm that reduces network energy consumption and minimizes data gathering delay is proposed. KDT selects the position of RPs by prioritizing the coverage of all sensors with reduced count of RPs and minimum RP coverage overlap. KD-tree-based strategy is utilized to efficiently query sensor node positions and enhance RP locations. Unlike many similar algorithms, the number of RPs are not predetermined; they are dynamically allocated based on the number and position of deployed sensor nodes such that a minimum number is selected that can provide full coverage of sensors. As such, the algorithm can adapt to any changes in network topology and modify the planned trajectory of the MS in response to sensor node failures. Furthermore, RPs are set to specific geographical locations rather than polling points to allow the MS to reach as many sensor nodes as possible in single hop communication, thus reducing the need for data forwarding and subsequently allowing sensors' energy to be conserved. Comparison of the discussed related works is summarized in Table 2.

3 Problem statement and system model

While several MS data gathering methods have been proposed in literature, these methods fall short on either data delivery time or communication energy consumption or both. In addition, the algorithm must be capable of adapting to network

Table 2 Comparison of the related works

Algo.	Objective	Approach	Metrics
Ref. [13]	Minimize latency	Rank-based multihop with MS	MS travel time
Ref. [14]	Latency minimization	Heuristic method	MS travel time and Network lifetime
Ref. [15]	MS path length and latency	Game theory and ACO	Energy consumption, data delivery delay, network lifetime
Ref. [16]	Three different path planning algorithms based on application requirements	Distributed approach	Energy consumption, delay and network lifetime
Ref. [17]	Minimize network energy consumption	Grid-based scheme	Energy efficiency and network lifetime
Ref. [18]	Optimize MS trip time	Cluster based, using ABC and ACO	Latency and energy efficiency
Ref. [19]	Solve hotspot problem	Iterative clustering method	Path length and energy efficiency
Ref. [20]	Data fusion and latency	Virtual grid-based clustering and neural network	Latency and network lifetime
Ref. [21]	Parking and moving data gathering by MS	Two optimization models under varying constraints	Data delivery delay
Ref. [22]	Path planning with coverage	PSO to select SPs and ACO to plan MS path	Network lifetime
Ref. [23]	RP selection and path planning	Directed spanning tree and ACO	Maximize network lifetime and Minimize delay
Ref. [24]	Extended the algorithm of [23]	selection of virtual RPs	Energy consumption and Lifetime
Ref. [25]	WSN data gathering with limited MS mobility	Heuristic algorithm, grid-based method	Energy efficiency
Ref. [26]	Optimize MS tour and balance the RPs' load	Path encoding method using multi-objective PSO	Energy, latency and lifetime
Ref. [27]	MS path planning	Convex polygon method	Data delivery efficiency and lifetime
Ref. [29]	Minimize data gathering delay	Optimizing the speed and data receiving schedule of the MS	Data delivery efficiency and latency
Proposed KDT scheme	Coverage aware and adaptive RP selection and MS path planning in response to node failures	KD-tree and ACO-based algorithm	Data delivery latency, Nodes' coverage, MS path length

topology changes and should allow modifying the planned MS trajectory in response to node failures. Hence, a KD-Tree-based scheme (KDT) is proposed to pinpoint the best locations for MS to stop and gather data in such a way that it reduces network energy consumption and minimizes data gathering delay. RPs are set to unique geographical positions to enable the MS to access as many sensor nodes as possible in a single hop communication, eliminating the need for data forwarding and, as a result, allowing sensors' resources to be conserved. The objective of the path planning algorithm is to plan a trajectory for the MS that allows it to collect data from most of the sensor nodes in single hop communication and thereby effectively reducing the network energy consumption. Unlike many existing algorithms, the number of RPs are not predetermined; they are dynamically allocated based on the number and location of deployed nodes. The algorithm can, thus, adapt to network topology changes and modify the planned MS trajectory in response to node failures. The planned trajectory can be updated to account for any nodes that fail due to energy depletion or other factors. These nodes should be removed from the planned path so that the total path distance can be reduced and data gathering delay minimized. Moreover, positioning of RPs focus to prioritize coverage of all sensors. Overlapping coverage of RPs is minimized, while guaranteeing 100% coverage of sensors by RPs. The shortest tour through the RPs is found through Ant Colony Optimization (ACO).

The network model consists of a square sensing region in which N sensor nodes are deployed for continuous monitoring. Their communication range (R_c) is double their sensing radius (R_s). It is assumed that all nodes are aware of their location and the location of their one-hop neighbors via GPS or some other localization approaches [30, 31]. Nodes are initially deployed in such a way that they cover the entire region of interest [32, 33]. Sensor nodes have a limited amount of energy, and they fail to operate when their energy runs out. A mobile sink (MS) is employed to follow a calculated trajectory and collect data from the deployed sensor nodes. It moves at a fixed velocity (v) and stops at specified locations in the planned trajectory—called rendezvous points (RPs)—to communicate with the sensors in the vicinity. RPs are dynamically determined as needed based on sensor deployment. The sensors upload their sensing data, their battery level, and their location. The communication range of the MS is the same as that of the deployed sensors.

4 Proposed KDT scheme

In this section, the path planning algorithm (KDT) that consists of two phases is discussed: (i) KD-Tree-based selection of rendezvous points (Sect. 4.1) and (ii) ACO-based route planning for MS (Sect. 4.2). The first phase is concerned with determining the best number and location of RPs. It relies on nearest neighbor and range searches with the help of a KD-Tree for efficient searching [11]. Ideally, the RP locations should allow the MS passing through these locations to communicate with the maximum number of sensor nodes in a single hop. The second phase aims to construct an efficient tour for the MS, whereby it can visit all the assigned RPs in

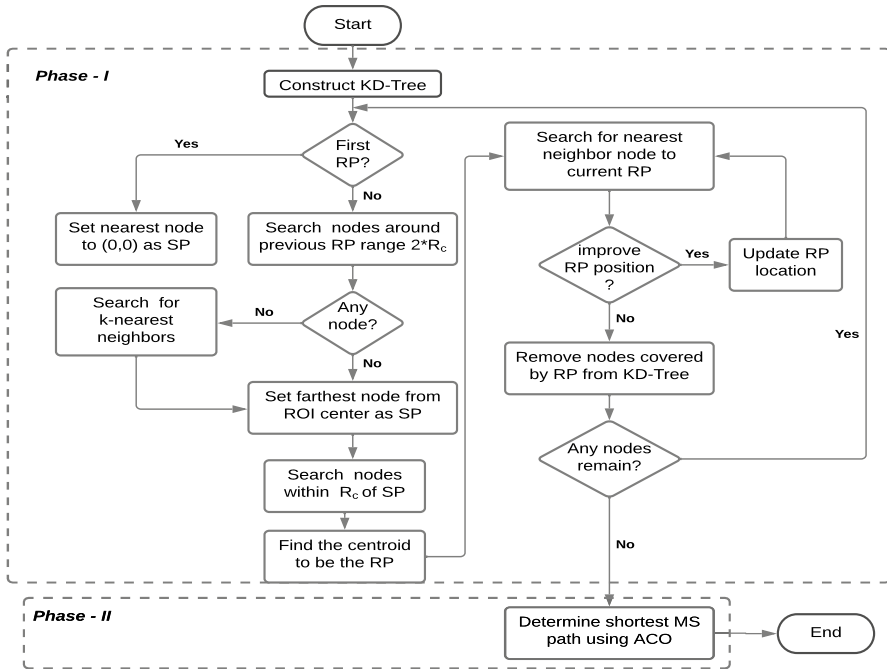


Fig. 1 Flow diagram of proposed KDT algorithm

the shortest time possible. Ant Colony Optimization (ACO) is utilized to find the shortest path through RPs [34]. Figure 1 shows the flowchart of the KDT algorithm.

4.1 KD tree-based RP selection (Phase-I)

In this phase, constructed KD-tree structure is utilized to determine the best location for the RPs. As shown in Fig. 2, the problem of RP positioning is considered as a coverage problem in order to maximize the number of sensors enclosed by the radius of a RP. This radius represents the communication range of the MS when it is positioned at that RP's location. Therefore, any sensor node that lies inside of an RP radius will be able to communicate with the MS in a single hop at some point during the tour. On the other hand, it is required to consider that for each RP, the MS must pause to wait for data uploads, which means that as the number of RPs increases, the data gathering delay will also increase. Therefore, it is desirable to utilize the minimum number of RPs that still ensures all sensor nodes fall within a RP radius. The basic steps of Phase-I are summarized in Algorithm 1.

The first step is to construct the KD-tree to enable efficient searching of the deployed sensor nodes. Each sensor node location is considered as one data point. Figure 3 shows an example of a KD-tree constructed based on the deployed sensors' locations. This data structure can be used to speed up queries for k-nearest neighbors or even searches within a specific range. The KD-tree is constructed recursively by

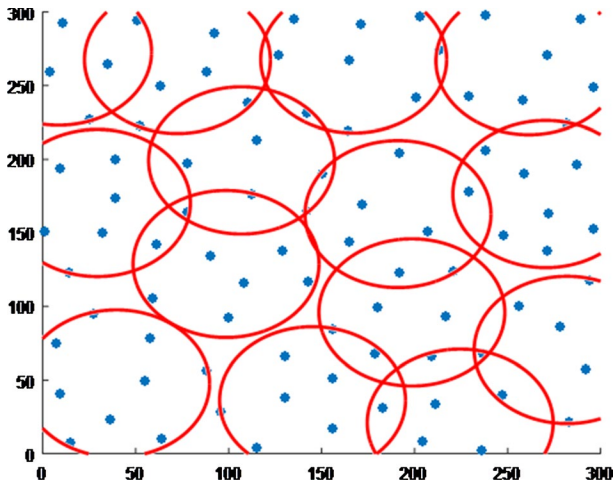
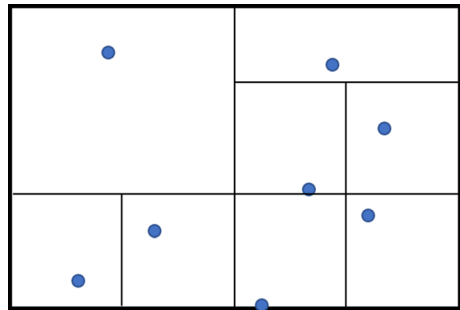


Fig. 2 RP positioning posed as coverage problem (blue dots represent sensor nodes, and the large red circles represent RP) (color figure online)

Fig. 3 KD-tree dividing the sensor field based on sensor node locations



continually splitting a node into two children and allocating the points to the appropriate children [11, 35]. Once the KD-tree is constructed, queries can be used to find the most suitable RP locations like those shown in Fig. 2. Figure 2 illustrates the RP positioning (posed as coverage problem) with blue dots representing the sensor nodes, and the large red circles representing the RP coverage.

The best location for each RP is determined by a looping algorithm. To start each round, a starting point (SP) is selected. For the first RP in the algorithm, the sensor node (SN) that is closest to the origin point (0, 0) is considered as the starting point. The selected origin point is the bottom-left corner of the region of interest (ROI). If it is not the first RP in the algorithm, then the SP is determined by starting with a range search query. Then data points within range of the previous RP center is checked, where the range is set to $2R_c$. However, if there are no points within this range, then the query is replaced with a nearest neighbor search to find a small number of neighbors. Whichever query is used, the result is a set of candidates for a SP. From these candidates, farthest from the ROI center is

selected to be the SP for the next round; that is, the candidate whose Euclidean distance from the ROI center is maximum. This is to ensure that RPs are created on the edge of the ROI first, which helps in preventing outliers as the algorithm progresses.

Algorithm 1 : KD-Tree based RP selection (Phase- I)

Input: Sensor node locations.

Output: Best position of RPs.

```

1: Construct KD-tree based on sensor node locations.
2: while tree is non-empty do
3:   if this is the first RP then
4:     search for the node closest to origin point (0,0) and denote it SP.
5:   else
6:     Search for nodes around previous RP in a range of  $2R_c$ .
7:     if no nodes are found then
8:       Search for  $k$  nearest neighbors.
9:     end if
10:    Select the node farthest from the ROI center to be the SP.
11:   end if
12:   Search for all nodes within communication range ( $R_c$ ) of the SP.
13:   Calculate the centroid of these and denote it as a RP.
14:   while RP position can be modified do
15:     Search for the nearest neighboring node to current RP.
16:     Check if RP can be moved to cover the nearest node while keeping all previous
17:     nodes covered.
18:     if yes then
19:       Move RP position to the new location.
20:     else
21:       Break loop.
22:     end if
23:   end while
24:   Remove nodes that are covered by the RP from the KD-tree.
25:   if no nodes remain in the tree then
26:     Break loop.
27:   end if
28: end while

```

Once the SP is determined, all nodes within the radius R_c are queried in order to calculate the centroid of the resulting nodes via Eq. (1), where k is the number of data points within the radius R_c , and x_k and y_k are the coordinates of the k -th point.

$$\left(C_x = \frac{x_1 + x_2 + \dots + x_k}{k}, C_y = \frac{y_1 + y_2 + \dots + y_k}{k} \right) \quad (1)$$

The resulting centroid is the tentative RP location which is illustrated in Fig. 4a. To enhance this location, repeated search for the closest sensor nodes that aren't covered by this RP is performed. If the RP location can be modified to cover an extra sensor node while also maintaining coverage of the previous nodes, then it is moved to the new location (Fig. 4b).

To do this, it is required to calculate the distance and direction in which to move the RP center. First, the nearest neighboring node that is not covered by the

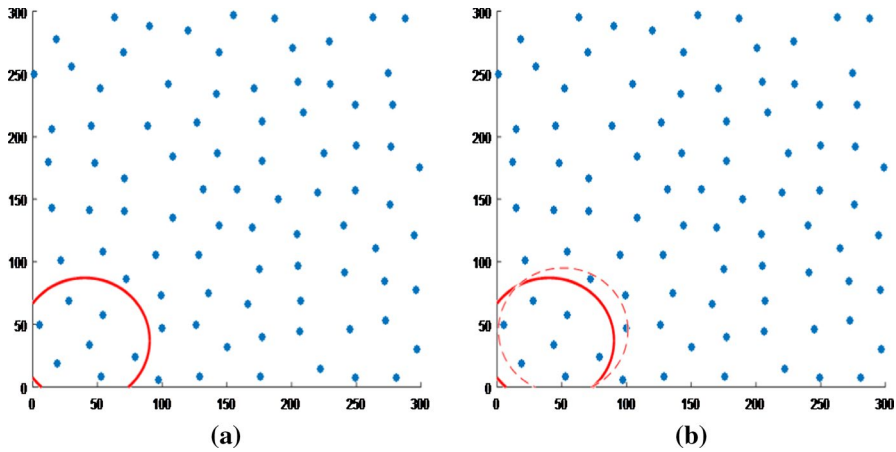


Fig. 4 **a** Tentative RP location, **b** modified location on the right, (blue dots represent sensor nodes, and the large red circles represent RP) (color figure online)

current RP is queried; it’s coordinates will be denoted (x_{nm}, y_{nm}) . Then, the direction vector (v_{dir}) is computed by Eq. (2), where (x_c, y_c) denotes the current RP center.

$$v_{dir} = \frac{\begin{bmatrix} x_{nm} - x_c \\ y_{nm} - y_c \end{bmatrix}}{\| \begin{bmatrix} x_{nm} - x_c \\ y_{nm} - y_c \end{bmatrix} \|} \tag{2}$$

Next, the allowable distance is determined to move in the given direction that will ensure that the sensor nodes already within the current RP are not uncovered. This is done by first finding the distance from every covered SN within the current RP to the RP radius edge in the opposite direction of v_{dir} . The minimum of these is the allowable distance d_a . Finally, the moving distance d_{mov} is found by Eq. (3).

$$d_{mov} = \left(\sqrt{(x_{nm} - x_c)^2 + (y_{nm} - y_c)^2} \right) - R_c \tag{3}$$

If $d_{mov} < d_a$, then the RP center location is moved with distance d_{mov} and direction v_{dir} . When the RP location cannot be moved further, the location is considered to be the best. After that, the sensor nodes within the current RP are removed from all future queries and the algorithm starts the next round to find the next RP. Figure 5a shows the RP locations without enhancement, and Fig. 5b shows the RP locations after enhancement.

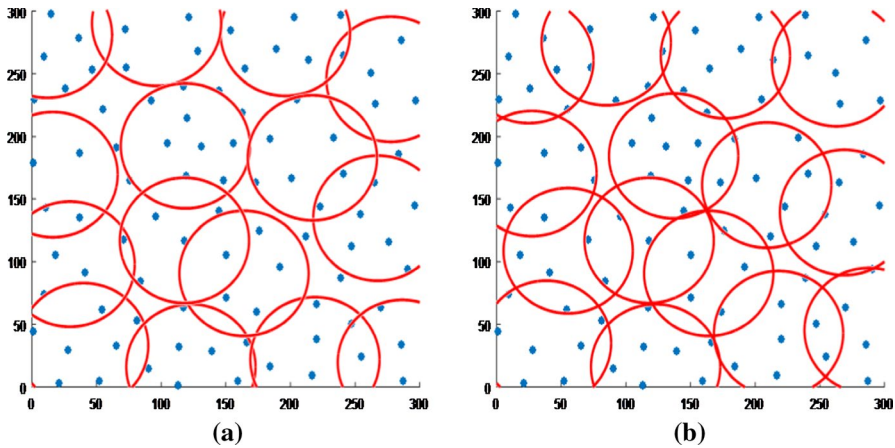


Fig. 5 RP locations with no enhancement are shown in **a**. Enhanced RP locations are shown in **b** (blue dots represent sensor nodes, and the large red circles represent RP) (color figure online)

4.2 ACO-based route planning for MS (Phase-II)

After the best RP locations are determined, the shortest route to visit each location needs to be determined. The MS will visit each RP center by following the designated route and pausing to collect data from the sensor nodes at each location. Sensor nodes will upload their sensing data as well as their battery level. Then, the MS returns to the base station to upload the gathered data and to receive an updated route for the next round if needed.

ACO is an established algorithm well-known to be suited for the Traveling Salesman Problem, which is defined as finding the shortest closed path through a given set of data points [36]. The set of RP centers can be modeled as a fully connected undirected graph and is denoted as $G = \langle V, E \rangle$; where V is the set of vertices representing the RP centers, and E is the set of edges connecting them. These edges are weighted by the Euclidean distance between the vertices that they connect. The basic steps of Phase-II are summarized in Algorithm 2. ACO is executed in the following steps:

1. The starting location of m ants is initialized randomly to be at any of the n available RPs. For each edge (i, j) , initialize its pheromone concentration $\tau_{ij}(t)$ to be a small positive constant. $\Delta\tau_{ij}$ is initialized to zero.
2. Each ant selects the next RP to visit based on edges' distance and pheromone concentration. The ant cannot select an RP that it has already visited. Selecting a particular edge, (i, j) has a probability $p_{ij}^k(t)$ calculated by Eq. (4). τ_{ij} represents the pheromone concentration on that edge. η_{ij} represents the destination visibility for the current ant, and it is calculated by the reciprocal of distance between RP_i and RP_j . α is a parameter that tunes the importance of the pheromone concentration, and β is a parameter that tunes visibility. Meanwhile, the set $allowed_k$ represents the RPs that have not yet been visited by the k -th ant.

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}(t)]^\alpha [\eta_{ik}]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

3. After moving to the chosen destination, the current tour length is updated by Eq. (5), where l_{ij} is the distance between RP_i and RP_j . b_{ij} is a Boolean described by Eq. (6).

$$L_k = \sum_{i,j \in RP, i \neq j} l_{ij} b_{ij} \tag{5}$$

$$b_{ij} = \begin{cases} 1 & \text{if } k\text{-th ant has used edge } (i,j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

4. The pheromone concentrations are all updated when a tour is completed. Any edges (i, j) that have been visited by the k -th ant in its tour are updated according to Eqs. (7)–(9); in which ρ controls the evaporation rate of pheromones on the trail, Q is a constant, and $\Delta\tau_{ij}^k$ is the pheromone concentration laid by k -th ant on edge (i, j) between time t and $(t + 1)$.

$$\tau_{ij}(t + 1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \tag{7}$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \tag{8}$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } k\text{-th ant used } (i,j) \text{ between } t \text{ and } t + 1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

5. Steps 1–4 are repeated until the maximum number of iterations is reached.

Algorithm 2 : ACO based MS route planning (Phase- II)

Input: RP locations.

Output: Best path for the mobile sink.

- 1: **while** path is being constructed **do**
 - 2: Initialize the ant positions at any of the RPs.
 - 3: **for** each ant **do**
 - 4: Determine the next RP to visit
 - 5: Update the current tour length
 - 6: Repeat until ant has finished a tour
 - 7: **end for**
 - 8: Update the pheromone concentration
 - 9: **if** maximum iterations reached **then**
 - 10: Break loop.
 - 11: **end if**
 - 12: **end while**
-

5 Simulation results and analysis

In this section, the performance of KDT is evaluated in terms of number of rendezvous points per tour, data delivery delay, and energy consumption. The results are compared with the following MS based data gathering algorithms: TRP-MC [22] and CTOS [29]. The following subsections discuss the simulation environment and experimental results.

5.1 Simulation environment

The simulations presented in this paper were implemented in MATLAB R2015a. A 300×300 m square region of interest is considered, and the number of nodes is taken in the range of 90–300. The sensing radius of the sensors is 25 m, and their communication radius is 50 m. The communication radius of the MS is taken as 50 m. The MS moves with a fixed velocity of 2 m/s and stops at each RP for 5 s. According to the distance between the transmitter and the receiver, the energy consumption is measured by radio energy dissipation model [16]. The energy and communication models adopted in this paper are similar to Alsaafin et al. [16]. The energy consumed by the transmitter or receiver electronic circuitry (E_e) is given by 50 nJ/bit while that of the transmitter amplifier (E_a) is 100 pJ/bit/m². To make the comparison better, other parameters used are similar to that of Gao et al. [22] and Dash et al. [29]. Different parameters used in the study are shown in Table 3.

5.2 Simulation results

The results of the simulation experiments are evaluated based on different performance metrics as given below.

- Number of selected RPs: this metric evaluates the count of RPs selected for planning the MS path under different algorithms
- Overlap in RP coverage: it is the measure of overlap in coverage of RPs (or redundant coverage)
- Ratio of sensors covered by at least one RP: the number of sensors covered by RPs directly impacts the energy consumed by sensors for communication purposes. This metric evaluates the ratio of covered sensors by the RPs.
- MS travel distance: gives the total distance of the MS' planned route for a varying number of sensors.
- Total trip time of the MS: this evaluates the time taken by the MS to complete its trip, which is influenced by two major factors: traveling distance and number of RPs.

Table 3 Various parameters and their values

Parameter	Value
ROI	$300 \times 300 \text{ m}^2$
Number of nodes	90–300
Sensing range	25 m
Communication range	50 m
Communication range of MC	50 m
Velocity of MCs	2 ms^{-1}
MC pause-time at PP	5 s

- Total energy consumption: it is the cumulative energy consumed by sensors in communication.

5.2.1 Comparison: number of selected RPs

Figure 6 shows the number of rendezvous points that the MS will visit in order to gather data from the sensors. A lower number of RPs is preferable because fewer stop points means a shorter travel route and lower data gathering delay. For TRP-MC, the number of RPs is predetermined at network startup and remains constant for the rest of the network's lifetime. Meanwhile, both CTOS and KDT have a dynamic number of RPs that changes based on the number of sensors in the field. For both algorithms, the number of RPs steadily decreases as the number of sensors decrease. However, CTOS creates many more RPs than the other algorithms for the same number of deployed sensors. This is because CTOS selects RPs based on intersecting regions of the sensors' communication ranges. Furthermore, there is no functionality for adjusting RP positions, and therefore, more RPs are needed to achieve full coverage of all the sensors.

5.2.2 Comparison: overlap in RP coverage

Figure 7 illustrates the overlap in RP coverage. Ideally, RP positions should be selected such that they can achieve full coverage of all deployed sensors while minimizing the overlap between RPs. This is because the RPs are the locations where the MS will stop and wait for sensors to upload their data. If there is overlap in RP coverage of the sensors, then their positions are not the best because there exists redundant coverage of the sensors. This can lead to a longer travel path and higher data gathering delay.

From Fig. 7, it is clear that TRP-MC has the lowest overlap ratio. It uses Particle Swarm Optimization to optimize the locations of the RPs and minimize their overlap. However, there is a trade-off between overlap and sensor coverage. Figure 8 shows that, by prioritizing overlap minimization in TRP-MC, sensor coverage suffers as a result. Meanwhile, CTOS does not consider RP overlap at all, and the overlap ratio goes as high as 80%, which is extremely redundant.

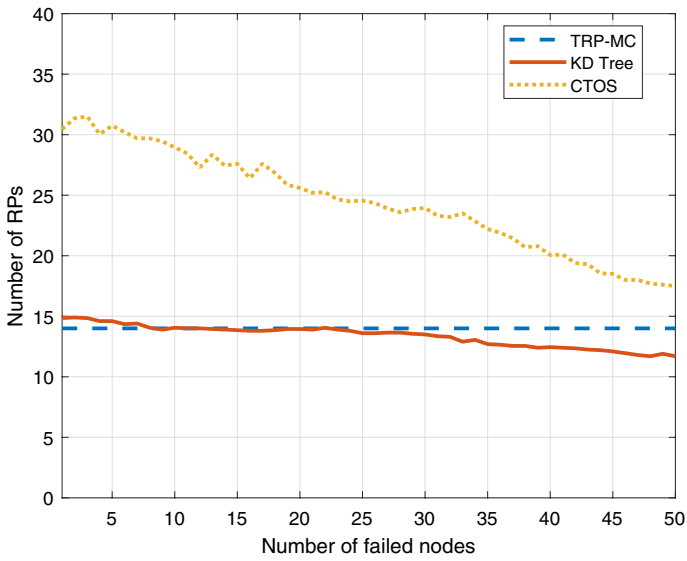


Fig. 6 Number of rendezvous points

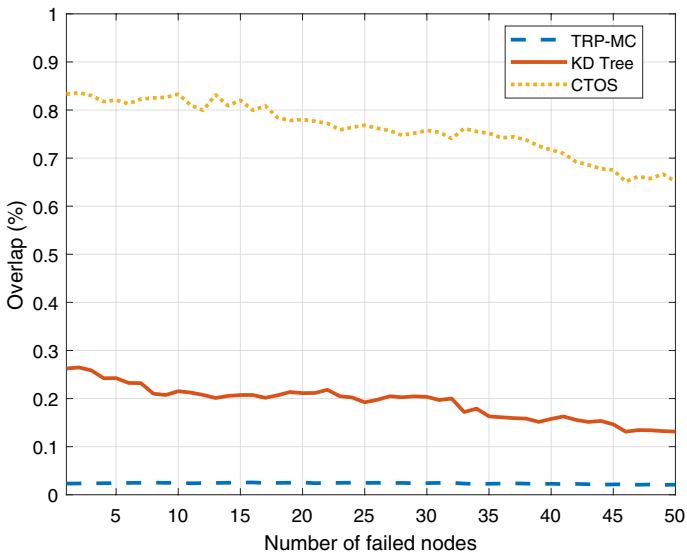


Fig. 7 Overlap ratio of rendezvous point coverage

KDT aims to minimize overlap while guaranteeing full sensor coverage. Therefore, the RP overlap of KDT may go up to 20% depending on the number of deployed sensors.

5.2.3 Comparison: ratio of sensors covered by at least one RP

The number of sensors covered by RPs directly impacts the energy consumed by sensors for communication purposes. If a sensor is covered by a RP, that means it can directly upload its data to the MS using single-hop communication. However, if a sensor is not covered by an RP, then it cannot upload its data directly, and must rely on a neighboring sensor node to forward its data to the MS. This is called multi-hop communication, and it causes the overall energy consumption in the network to increase.

As seen in the previous figure, TRP-MC prioritizes minimizing overlap between RPs which causes several sensor nodes to be left uncovered by a RP. TRP-MC achieves a maximum coverage of 80% of deployed sensors. The rest of the sensors must resort to multi-hop communication to get their data uploaded to the MS, which, in turn causes network energy consumption to increase. Meanwhile, CTOS and KDT position RPs to prioritize coverage of all sensors, if possible. Their performance is almost identical in terms of sensor coverage. However, as seen before, CTOS employs twice as many RPs to achieve the same result; and this has a detrimental effect on travel route length and data gathering time which is shown in the next experiments.

5.2.4 Comparison: MS travel distance

Figure 9 illustrates the total distance of the MS' planned route for a varying number of sensors. It is important to minimize the planned path length when possible in order to minimize data delivery delay. TRP-MC calculates a path only once at

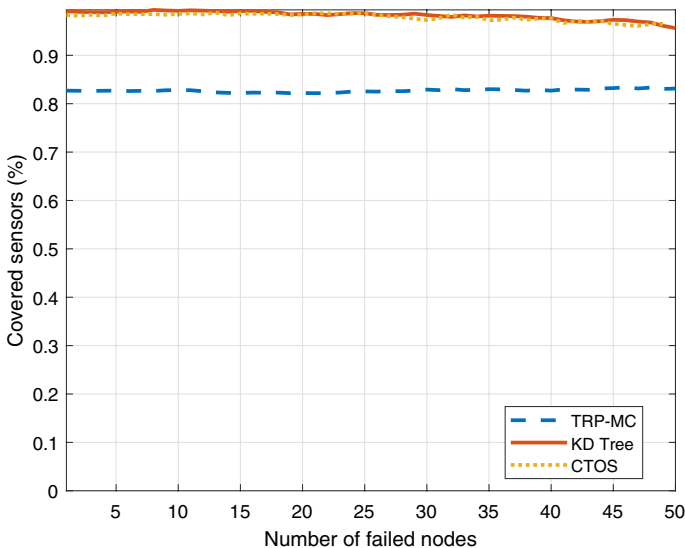


Fig. 8 Ratio of sensors covered by at least one rendezvous point

network startup; therefore, the path length is steady regardless of the number of sensors. CTOS and KDT dynamically select a path based on the number and position of the sensors. As a further analysis, the number of nodes is varied from 90 to 300, and the MS travel distance is examined. The results are depicted in Fig. 10. As evident from the results, out of the three algorithms, KDT has the best performance, followed by CTOS, and finally, TRP-MC.

5.2.5 Comparison: total trip time of the MS

The total trip time of the MS for different algorithms are shown in Fig. 11. It is affected by two major factors: traveling distance and number of RPs (Figs. 9 and 6, respectively). The MS must travel the whole distance at a certain velocity, and it stops at every RP to wait for data to be uploaded by the sensors. TRP-MC and KDT start with a similar number of RPs, but KDT has a shorter traveling distance. Furthermore, for KDT, the traveling distance and number of RPs continue to decrease as the number of sensors drop. Therefore, the total trip time steadily decreases for KDT as opposed to TRP-MC, which remains the same throughout the lifetime of the network. For CTOS, the number of RPs is extremely high, and the MS must stop at each one for the specified interval, which means that CTOS achieves the worst performance of the three algorithms in terms of data delivery delay.

5.2.6 Comparison: total energy consumption

Figure 12 shows the total cumulative energy used by sensors for communication. Sensors covered by RPs can communicate directly with the MS in a single hop

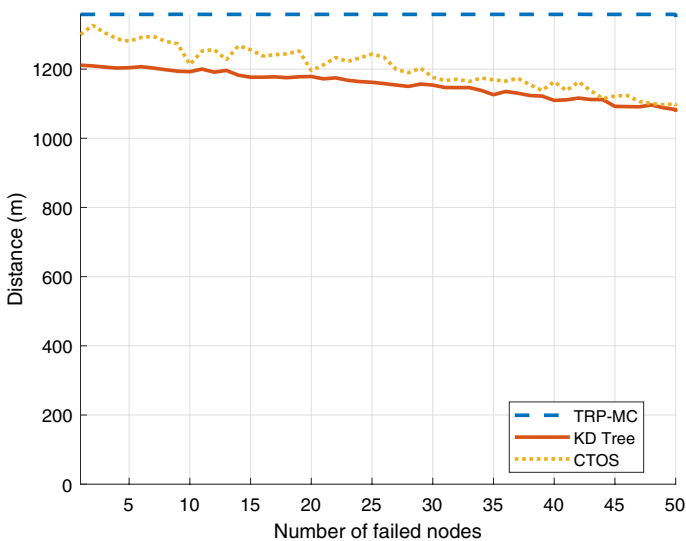


Fig. 9 Travel distance of the mobile sink

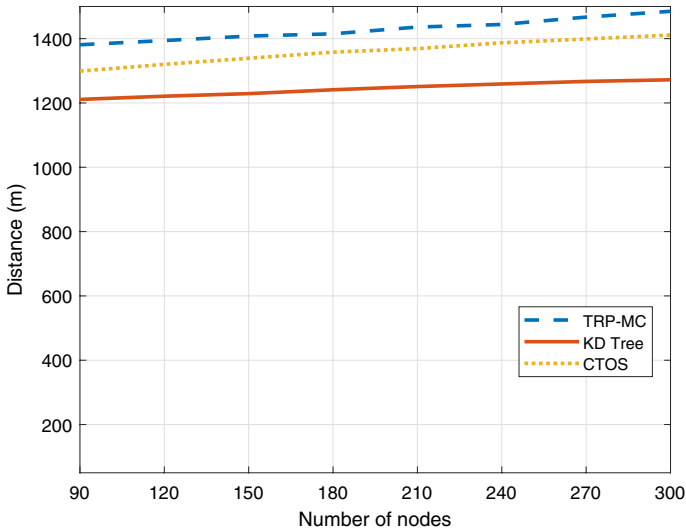


Fig. 10 Travel distance of the mobile sink with varying number of nodes

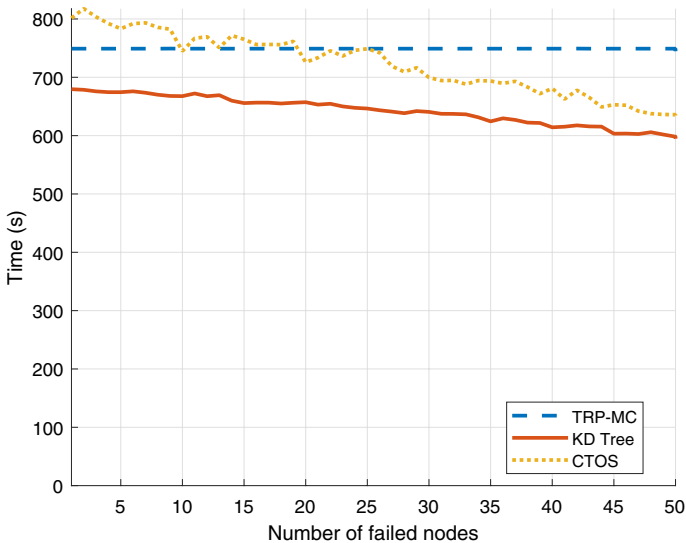


Fig. 11 Trip time of the mobile sink

which minimizes energy consumption and prolongs the network lifetime. TRP-MC has the highest total energy consumption because many sensors must resort to multi-hop communication due to being left out of RP coverage. Meanwhile, CTOS and KDT maintain close to 100% coverage, and therefore, their energy consumption is more efficient. However, from the previous figures, it can be concluded that KDT

achieves high energy efficiency, while dramatically reducing the data delivery delay compared to similar algorithms.

5.2.7 Performance analysis

Path planning for data gathering in WSN with MS is an important and challenging concern. The network's performance is heavily influenced by the scheduled path. If the planned path is too lengthy, the MS will take a long time to travel, and the network will suffer from significant network latency. Many sensors may not interact directly with the mobile collector if the planned path is too short, wasting a lot of energy for data forwarding. As a result, scheduling a short path while ensuring better coverage is challenging. The proposed KD-Tree-Based Scheme (KDT) not only reduces energy consumption and latency but also is robust and adaptive to network topology changes, and the planned MS trajectory can be refined to accommodate node failures. A lower number of RPs is preferable because fewer stop points means a shorter travel route and lower data gathering delay. The number of RPs in TRP-MC is specified at network initiation and remains consistent during the network's existence. Meanwhile, CTOS and KDT both feature a dynamic number of RPs that varies depending on the number of sensors in the field. The number of RPs for both methods progressively reduces as the number of sensors decreases. For the same number of deployed sensors, CTOS generates considerably more RPs than the other algorithms. This is because CTOS chooses RPs depending on the intersecting communication ranges of the sensors. Furthermore, there is no way to change the location of the RPs. Thus, more RPs are required to cover all of the sensors completely. TRP-MC achieves a maximum coverage of 80% of deployed sensors. The rest of the sensors must resort

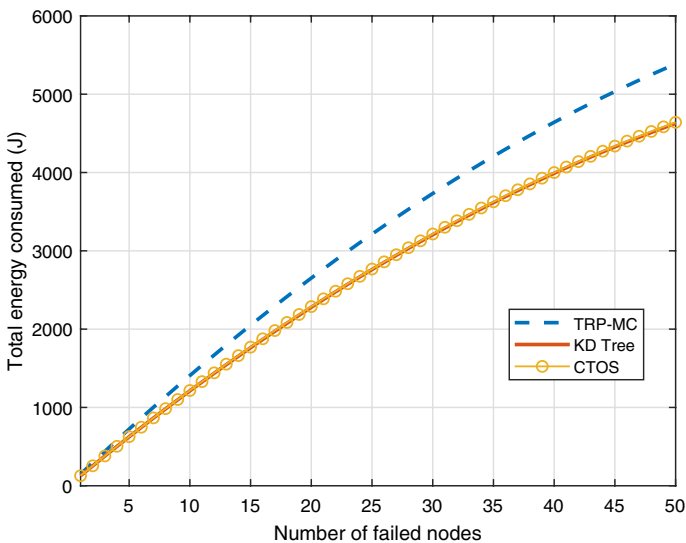


Fig. 12 Cumulative energy consumed by sensors in communication

to multi-hop communication to get their data uploaded to the MS, which, in turn causes network energy consumption to increase. Meanwhile, CTOS and KDT position RPs to prioritize coverage of all sensors, if possible. Their performance is almost identical in terms of sensor coverage (Fig. 8). However, as seen before, CTOS employs twice as many RPs to achieve the same result (Fig. 6), which has a negative impact on path length and data gathering time which is shown in the subsequent experiments (Figs. 9, 11).

6 Conclusion

Efficient path planning plays a significant role in WSN performance. In this paper, the best locations for the planned path's RPs are determined by treating it as a coverage problem. A KD-tree is built based on node locations, which can then be used for efficient searching. The number of rendezvous points is determined dynamically by the algorithm such that a minimum number is selected that can provide 100% coverage of sensors. Once the number and location of RPs is selected, the ACO algorithm is used to connect all RPs in a shortest path loop. The MS traverses this path and gathers data from all sensor nodes in single hop communications. Simulations show that KDT consistently provides the MS with an effective route that can cover all sensors in its communication path. This means that KDT achieves high energy efficiency for the sensors and reduces data delivery delay when compared to recent algorithms.

In future, the work is planned to be extended to develop new algorithms for very large-scale network and also consider non-uniform data constraints of nodes, by utilizing multiple mobile sinks. The algorithm will also be analyzed by considering non-homogeneous sensor nodes in future works. In addition, different machine learning models are planned to be analyzed to further optimize the algorithm performance in data gathering and the MS traveling path selection.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Al Aghbari Z, Khedr AM, Osamy W, Arif I, Agrawal DP (2020) Routing in wireless sensor networks using optimization techniques: a survey. *Wirel Pers Commun* 111(4):2407–2434
2. Abu Safia A, Al Aghbari Z, Kamel I (2016) Phenomena detection in mobile wireless sensor networks. *J Netw Syst Manag* 24(1):92–115
3. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. *Comput Netw* 52(12):2292–2330
4. Osamy W, Khedr AM, Aziz A, El-Sawy AA (2018) Cluster-tree routing based entropy scheme for data gathering in wireless sensor networks. *IEEE Access* 6:77372–77387

5. Osamy W, El-sawy AA, Khedr AM (2019) Satc: a simulated annealing based tree construction and scheduling algorithm for minimizing aggregation time in wireless sensor networks. *Wirel Pers Commun* 108(2):921–938
6. Khedr AM (2015) Effective data acquisition protocol for multi-hop heterogeneous wireless sensor networks using compressive sensing. *Algorithms* 8(4):910–928
7. Ahmed N, Kanhere SS, Jha S (2005) The holes problem in wireless sensor networks: a survey. *ACM SIGMOBILE Mob Comput Commun Rev* 9(2):4–18
8. Omar DM, Khedr AM, Agrawal DP (2017) Optimized clustering protocol for balancing energy in wireless sensor networks. *Int J Commun Netw Inf Secur* 9(3):367–375
9. Mehto A, Tapaswi S, Pattanaik K (2020) A review on rendezvous based data acquisition methods in wireless sensor networks with mobile sink. *Wirel Netw* 26(4):2639–2663
10. Al Aghbari Z, Kamel I, Elbaroni W (2013) Energy-efficient distributed wireless sensor network scheme for cluster detection. *Int J Parallel, Emerg Distrib Syst* 28(1):1–28
11. Friedman JH, Bentley JL, Finkel RA (1977) An algorithm for finding best matches in logarithmic expected time. *ACM Trans Math Softw (TOMS)* 3(3):209–226
12. Bentley JL (1975) Multidimensional binary search trees used for associative searching. *Commun ACM* 18(9):509–517
13. Miao Y, Sun Z, Wang N, Cao Y, Cruickshank H (2016) Time efficient data collection with mobile sink and Ymimo technique in wireless sensor networks. *IEEE Syst J* 12(1):639–647
14. Tang J, Guo S, Yang Y (2015) Delivery latency minimization in wireless sensor networks with mobile sink. In: 2015 IEEE International Conference on Communications (ICC). IEEE, pp 6481–6486
15. Raj PP, Al Khedr AM, Aghbari Z (2020) Data gathering via mobile sink in WSNs using game theory and enhanced ant colony optimization. *Wirel Netw* 26(4):2983–2998
16. Alsaafin A, Al Khedr AM, Aghbari Z (2018) Distributed trajectory design for data gathering using mobile sink in wireless sensor networks. *AEU-Int J Electron Commun* 96:1–12
17. Majma MR, Almassi S, Shokrzadeh H (2016) Sgdd: self-managed grid-based data dissemination protocol for mobile sink in wireless sensor network. *Int J Commun Syst* 29(5):959–976
18. Wang Z, Ding H, Li B, Bao L, Yang Z (2020) An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. *IEEE Access* 8:133577–133596
19. Park J, Kim S, Youn J, Ahn S, Cho S (2020) Iterative sensor clustering and mobile sink trajectory optimization for wireless sensor network with nonuniform density. *Wire Commun Mobile Comput*
20. Wang J, Gao Y, Liu W, Sangaiah AK, Kim H-J (2019) An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks. *Int J Distrib Sens Netw* 15(3):1550147719839581
21. Han Z, Shi T, Lv X, Jia X, Wang Z, Zhou D (2019) Data gathering maximisation for wireless sensor networks with a mobile sink. *Int J Ad Hoc Ubiquitous Comput* 32(4):224–235
22. Gao Y, Wang J, Wu W, Sangaiah AK, Lim S-J (2019) Travel route planning with optimal coverage in difficult wireless sensor network environment. *Sensors* 19(8):1838
23. Kumar P, Amgoth T, Annavarapu CSR (2018) Aco-based mobile sink path determination for wireless sensor networks under non-uniform data constraints. *Appl Soft Comput* 69:528–540
24. Donta PK, Amgoth T, Annavarapu CSR (2021) An extended ACO-based mobile sink path determination in wireless sensor networks. *J Ambient Intell Human Comput* 12(10):8991–9006
25. Wu C, Liu Y, Wu F, Fan W, Tang B (2017) Graph-based data gathering scheme in WSNs with a mobility-constrained mobile sink. *IEEE Access* 5:19463–19477
26. He X, Fu X, Yang Y (2019) Energy-efficient trajectory planning algorithm based on multi-objective PSO for the mobile sink in wireless sensor networks. *IEEE Access* 7:176204–176217
27. Wen W, Zhao S, Shang C, Chang C-Y (2018) EAPC: energy-aware path construction for data collection using mobile sink in wireless sensor networks. *IEEE Sens J* 18(2):890–901
28. Anzola J, Pascual J, Gonzalez Tarazona G, Crespo R (2018) A clustering WSN routing protocol based on KD tree algorithm. *Sensors* 18(9):2899
29. Dash D, Kumar N, Ray PP, Kumar N (2020) Reducing data gathering delay for energy efficient wireless data collection by jointly optimizing path and speed of Mobile Sink. *IEEE Syst J* 15(3):3173–3184
30. Khalifa B, Al Aghbari Z, Khedr AM, Abawajy JH (2017) Coverage hole repair in WSNs using cascaded neighbor intervention. *IEEE Sens J* 17(21):7209–7216

31. Htun AM, Maw MS, Sasase I (2014) Reduced complexity on mobile sensor deployment and coverage hole healing by using adaptive threshold distance in hybrid wireless sensor networks. In: IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), vol 2014. IEEE, pp 1547–1552
32. Mini S, Udgata SK, Sabat SL (2014) Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sens J* 14(3):636–644
33. Sengupta S, Das S, Nasir M, Panigrahi BK (2013) Multi-objective node deployment in WSNs: in search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. *Eng Appl Artif Intell* 26(1):405–416
34. Dorigo M, Maniezzo V, Colorni A et al (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern B Cybern* 26(1):29–41
35. Merry B, Gain J, Marais P (2013) Accelerating kd-tree searches for all k-nearest neighbours. Tech. Rep., University of Cape Town
36. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1(1):53–66

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.