Journal of Ambient Intelligence and Humanized Computing (2021) 12:10837–10864
https://doi.org/10.1007/s12652-020-02886-z

ORIGINAL RESEARCH

An optimal mobile sink sojourn location discovery approach for the energy-constrained and delay-sensitive wireless sensor network

Saugata Roy[1] · Nabajyoti Mazumdar[2] · Rajendra Pamula[1]

Received: 16 March 2020 / Accepted: 23 December 2020 / Published online: 15 January 2021

Abstract
Latest studies have exploited sink mobility as a pervasive approach to effectively mitigate the energy hole problem frequently perceived in the clustered wireless sensor network. For mobile sink (MS) based data collection, the selection of MS sojourn location plays a significant role in network performance. Moreover, there are several challenges to WSN in the form of energy preservation and data delivery latency. A one-hop data collection by the MS from each cluster conserves the network energy significantly but increases the number of sojourn points (SPs) which elongates the MS tour time. On the contrary, a lesser number of SPs may indulge the long chain of multi-hop data forwarding, which may further bring the energy hole issue into existence. In order to find a trade-off between energy preservation and data delivery delay, this article aims to attain an optimal number of SPs that guarantee the one-hop data collection from each cluster. Subsequently, an MS traversal path is formed by exploiting the ant colony optimization (ACO) algorithm that generates a near-optimal solution. Simulation results manifest the efficacy of the proposed protocol over the existing ones in terms of network lifetime, energy consumption, packet delivery ratio, end-to-end delay, etc.

Keywords Ant colony optimization · Energy hole · Highest density node · Mobile sink · Sojourn point · Wireless sensor network

1 Introduction

1.1 Background

Since the last decade, the notion of clustering (Shahraki et al. 2020; Heinzelman et al. 2000; Akkari et al. 2015) has been widely embraced in the domain of wireless sensor network (WSN) to diminish the data redundancy among the sensed data. It also simplifies the routing process as only the cluster heads (CHs) are allowed to engage in routing. However, clustering, along with multi-hop routing (Yu et al. 2012; Sabet and Naji 2015; Alaei and Yazdanpanah 2019; Osamy et al. 2018; Salem and Shudifat 2019) triggers rapid energy depletion of the sensor nodes in close proximity to the sink that accelerates their premature death. This leads to the isolation of the sink from the rest of the network, which is popularly known as energy hole problem (Ren et al. 2015; Ramos et al. 2016). Energy hole is one of the critical issues that causes a deterioration in the quality of service (QoS) of WSN by reducing the packet delivery ratio, throughput, network lifetime, and so on. It has been observed from most state-of-the-art protocols (Tao et al. 2012; Tang et al. 2015; Ghafoor et al. 2014; Roy et al. 2020; Abo-Zahhad et al. 2015; Alsaafin et al. 2018; Wen et al. 2017b; Mehto et al. 2020; He et al. 2019; Krishnan et al. 2018a; Kumar et al. 2018; Wen et al. 2017a; Krishnan et al. 2018b; Huang et al. 2019; Wen et al. 2018; Huang et al. 2017) that exploitation of mobile sink (MS) for sensor data gathering has turned out as a promising approach to competently alleviate the energy hole issue. Generally, in a clustered WSN, MS is a movable device that is responsible for gathering the cluster data from the CHs by travelling throughout the deployment area and eventually supplying the gathered data to the base station (BS). Sink mobilization assures the hotspots around

✉ Saugata Roy
saugataroy15@gmail.com

[1] Department of Computer Science and Engineering, Indian Institute of Technology (ISM), Dhanbad, India

[2] Department of Information Technology, Central Institute of Technology, Kokrajhar, India

the sink to be changed as the data routing load is evenly diffused throughout the network. Thus, the overall network energy is balanced, and its lifetime is prolonged.

As stated by the radio energy dissipation model (Heinzelman et al. 2000), energy consumption is directly proportional to the transmission distance, and when the transmission distance becomes greater than some threshold distance (δ_{th}), it raises the dissipated energy exponentially (following the multipath model). Hence, for direct transmission to the MS, it is desirable to limit the communication range of the CHs within δ_{th}, and consequently follow the free space model in order to conserve the network energy. In other words, if an MS reaches within δ_{th} distance of any CH, it can directly transmit (free space/one-hop delivery) the data; otherwise, it will obey the multi-hop communication until its data reaches to the MS. It is worth noting that the rest part of this paper will use the terms one-hop distance and threshold distance (δth) interchangeably during the MS-based data gathering. Usually, traditional MS-based data gathering approaches can be categorized into two major groups. In the first one, MS fetches data from individual sensor nodes via one-hop communication (Tao et al. 2012; Tang et al. 2015), which results in improved network energy preservation and collision-free data collection. Nevertheless, this technique can not satisfy the requirements of delay-sensitive applications of WSN due to a large number of data collection points as well as the velocity constraint of the MS. The other school of thought (Ghafoor et al. 2014; Roy et al. 2020; Abo-Zahhad et al. 2015; Alsaafin et al. 2018; Wen et al. 2017b; Mehto et al. 2020; He et al. 2019; Krishnan et al. 2018a; Kumar et al. 2018; Wen et al. 2017a; Krishnan et al. 2018b) allows the MS to sojourn at a limited number of data collection points for data gathering and thus reduces the data delivery latency to the BS. Such a delay-sensitive data gathering approach is often termed as rendezvous based strategy where some of the sensor nodes are elected as rendezvous points (RPs). The non-RP nodes forward their data to the nearest RPs that eventually deliver these buffered data to the MS when it appears in their proximity. Determining the appropriate RPs, along with an optimal MS trajectory, is the utmost attention of this strategy. However, most of the existing rendezvous based approaches do not ensure a one-hop data gathering because of the restricted number of MS sojourn points (SPs); in consequence, it may encounter multi-hop communication. This raises the probability of the energy hole around the SPs. Considering the incompatibility between energy preservation and data delivery delay, MS-based data gathering protocols should exploit the advantages of both delay and energy-sensitive approaches. This arises the research problem of finding a minimal set of SPs to secure one-hop data delivery; accordingly, the trade-off between the aforementioned conflicting metrics can be achieved.

Once the SPs are nominated, it is necessary to design an efficient traversal path for the MS to visit each SP for data collection. Generally, MS traversal path designing technique aims to minimize the length of the path. Researchers in (Wen et al. 2017b; Mehto et al. 2020; He et al. 2019; Krishnan et al. 2018a; Kumar et al. 2018) have concluded that an optimal tour of MS not only reduces the data collection latency but also improves the packet delivery ratio. However, finding the minimal length MS tour among all the possible tour sequences incurs a non-polynomial time solution for a large number of SPs that refers to the travelling salesman problem (TSP) (Kumar et al. 2018). Since the past two decades, several nature-inspired metaheuristics such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), moth flame optimization, bee colony optimization etc. are leveraged by the researchers to generate a near-optimal solution of the TSP in a reasonable amount of time. However, finding the optimal MS tour problems are considered as a discrete optimization problem, and unlike GA and PSO, the aforesaid metaheuristics are mainly designed for continuous optimization. This prevents such metaheuristics from being suited for the MS traversal problems in WSN. On the other hand, between ACO and GA, ACO (Dorigo and Stützle 2019; Krishnan et al. 2019) is a construction based technique and more suitable for MS traversal while as a population based search heuristic, the working of GA increases the search overhead, and consequently not viable for MS based routing. In addition, the majority of the metaheuristics demand a proper tuning of the parameters to attain better efficacy that results in severe computational cost for a large-scale problem.

Unlike the other metaheuristics, ACO adopts an adaptive approach for parameter tuning during its search procedure. Considering the foregoing benefits, ACO has gained its popularity in solving the computationally hard routing problems like TSP and generates an optimal MS tour such that the path length is minimized while visiting all the nominated SPs (Krishnan et al. 2018a; Kumar et al. 2018).

1.2 Motivation

Although extensive research works have been carried out for decades on clustering in WSN, MS-based data gathering from a clustered WSN is yet to be explored particularly in its delay-sensitive applications. In the energy-constrained delay-sensitive network, MS-based data gathering reveals the crucial challenges as follows:

- Prolonging the network lifetime has always been the primary goal in an energy-constrained WSN. Most of the existing energy-sensitive MS based data gathering protocols achieve energy efficiency by ensuring one-hop connectivity among the CHs and SPs. However, they often follow visiting each CH for data gathering, which is not a scalable approach in large-scale networks. In this regard, the research problem of a minimal set of SP nomination emerges to guarantee every CH coverage by one-hop distance. A solution to such a problem will drastically improve the data collection latency while maintaining the energy efficiency of the network.

- Minimizing data collection latency is another key objective of WSN, especially for delay-sensitive applications. The data collection latency is directly related to the number of SPs of the MS. As a result, many of the existing delay-sensitive methods have nominated a restricted set of SPs to limit the tour time of the MS. However, limiting the number of SPs may result in multi-hop communication among the CHs and SPs that enhances the probability of energy hole problem. This necessitates the researchers to optimize the number of SPs that are surrounded by the maximum possible number of CHs.

Therefore, the above-mentioned challenges motivate us to design an optimal trajectory of the MS that passes through the minimum set of SPs while ensuring one-hop connectivity to all CHs.

1.3 Contribution

This article proposes an energy-constrained and delay-sensitive approach for discovering optimal sojourn points of the mobile sink. In a nutshell, the major contributions of this article are as follows:

- Nomination of the minimal number of SPs which guarantee the one-hop data collection from the CHs of the network.
- Construction of an optimal MS traversal path with the help of ant colony optimization algorithm to visit all the nominated SPs exactly once.
- Performing a detailed simulation analysis under (truncated) Gaussian, non-uniform, and uniformly distributed WSN to manifest the flexibility and improvement of the proposed algorithm over the existing ones.

To the best of our knowledge, this is the first attempt to form a delay-aware data gathering path for the MS by nominating a minimal set of SPs in a clustered WSN. It is to be noted that, since clustering in WSN is a well-explored topic, this article adopts any of the standard existing clustering algorithms (Yu et al. 2012; Sabet and Naji 2015) in order to form a clustered WSN.

1.4 Organization

The rest part of the paper is organized as follows. The related studies are reviewed in Sect. 2, while Sect. 3 consists of the system model comprising network deployment model, rudimentary assumptions, relevant terminologies, and problem formulation. The proposed methodology is vividly described in Sect. 4 while Sect. 5 presents a theoritical analysis of our methodoogy. Section 6 accomplishes a comparative experimental analysis of the proposed protocol with the related ones to show its efficacy. Finally, Sect. 7 draws the conclusion and future work.

2 Related work

Several studies on the cluster-based communication algorithms (Heinzelman et al. 2000; Akkari et al. 2015; Yu et al. 2012; Sabet and Naji 2015; Alaei and Yazdanpanah 2019; Osamy et al. 2018; Salem and Shudifat 2019) have been developed over the past 2 decades with the aim of minimizing data redundancy as well as energy consumption of the overall network. However, multi-hop routing based clustering scheme encounters energy hole problem, and at the outset, it was adequately alleviated by various unequal clustering strategies (Kaur and Kumar 2018; Hamidouche et al. 2018; Logambigai et al. 2018; Chen et al. 2019; Mazumdar and Om 2018; Sabor et al. 2016; Gajjar et al. 2016). Despite the fact, static sink scenario could not disseminate the data routing load uniformly in the network. Under the circumstances, sink mobility has emerged as a better alternative in order to competently mitigate the energy hole issue. Primarily, MS-based data gathering can be classified into two major classes.

2.1 Energy-sensitive approach

This class of data collection recommends the MS to reach individual sensor nodes for data collection; accordingly, it acquires an energy-saving architecture. Study Tao et al. (2012) develops a sink mobility assisted data gathering scheme based on progressive optimization method that considers the data rate constraints between MS and the static nodes. It aims to minimize the data gathering latency by combining the proximate collection sites, actively skipping the redundant nodes, and eventually finding the proper start and finish locations of data collection. In the article Tang et al. (2015), based on the travelling salesman problem (TSP) heuristic, the delivery latency minimization problem (DLMP) is studied in a randomly deployed WSN with mobile sink. The proposed substitution heuristic algorithm (SHA) not only finds the anchor points on the sensor nodes' communication radius but also removes the redundant anchor points, and consequently ensures better shortening of the MS travelling route. However, gathering the data from individual sensor nodes elongates the MDC travelling route, which may not fit the aforementioned studies in delay-sensitive applications of WSN.

2.2 Delay-sensitive approach

On the contrary, the second class of approach aims to produce lower data delivery latency. It suggests the MS visit only a limited set of sensor nodes referred as rendezvous points (RP) for data gathering within a delay bound while the non-RP nodes forward their sensory data to the nearest RPs (Roy et al. 2020; Abo-Zahhad et al. 2015; Alsaafin et al. 2018; Wen et al. 2017b; Mehto et al. 2020; He et al. 2019; Krishnan et al. 2018a; Kumar et al. 2018). Article Roy et al. (2020) proposes a distributed RP selection strategy that considers both the energy and coverage sensitive parameters for clustering and routing procedure. This kind of methodology guarantees minimal hop data forwarding from sensor nodes to their respective MS sojourn locations; accordingly, it attains a reducing data acquisition latency. A study in Abo-Zahhad et al. (2015) introduces an MS based energy-efficient clustering protocol MSIEEP that alleviates the energy hole problem. MSIEEP reduces the overall hop count of the network by dividing the sensing field into several equal-size partitions, each of which leads to one MS sojourn location. Furthermore, the proposed clustering protocol figures out the optimal number of cluster heads and their locations utilizing the adaptive immune algorithm (AIA). But the major pitfall of this policy is that it minimizes the average hop counts of the network successfully only when the number of sojourn locations is high, which may increase the data gathering latency. A further study Alsaafin et al. (2018) on MS trajectory planning for data gathering enables the network to dynamically pick up the RPs based on the clustering technique in a distributed way. This article recommends three types of MS trajectory designs referred to as REP, RDP, and DBP, which are respectively applicable for energy sensitive, delay-sensitive, and time-bounded applications of WSN.

Authors in Wen et al. (2017b) propose an energy-aware path construction (EAPC) algorithm whose notion is to consider the path cost between successive data collection points (DCP) for the purpose of acquiring an improved data collection path. EAPC algorithm is initiated by the building of the maximum spanning tree; then followed by the selection of an appropriate set of DCPs. Finally, it suggests the MS perform data collection from the highly burdened DCPs. Despite the improved data collection path construction, it may suffer from long haul multi-hop data forwarding in case of large scale unbalanced deployment scenarios. References Mehto et al. (2020); He et al. (2019) leverage particle swarm optimization (PSO) metaheuristic that aims to optimize the MDC travelling path in polynomial time. Study Mehto et al. (2020) presents an efficient RP nomination strategy utilizing the single objective PSO algorithm under two equality constraints, namely data delivery delay and traffic rate constraints. Authors in He et al. (2019) leverage multi-objective PSO based fitness function that aims to reduce the MDC tour length by an optimal selection of k number of MS sojourn points. However, in a large-scale WSN, it is inconvenient to forecast the suitable value of k as changes in the network topology may require a variation in the k value. Authors in Krishnan et al. (2018a) presents a dynamic clustering approach that distributes the dissipated energy load among the sensors by electing the CHs in each round; accordingly, it enhances the network lifetime. Subsequently, it establishes an optimal path for mobile data collectors by utilizing the ACO algorithm. However, it suffers from heavy congestion of the control messages due to which the network QoS may deteriorate. An ACO based routing algorithm is presented in the article (Kumar et al. 2018) that leverages the forwarding load of the sensor nodes to find the near-optimal set of RPs under uneven data generation.

From the literature analysis, it can be observed that the energy-efficient approaches overestimate the MS sojourn points by stopping the MS at each CH location to minimize the energy depletion. However, such approaches drastically elongate the tour time of the MS. On the other hand, the delay-sensitive approaches underestimate the MS sojourn points by nominating a limited SPs, which results in multi-hop data propagation among the CHs. Thus, there is a research gap observed to ensure the nomination of minimal SPs for CH data gathering while preserving free space connectivity for all CHs to any SP. In this context, this article proposes an optimal SP nomination strategy such that all the CHs are covered by one-hop distance with minimal SPs. Later, an optimal tour for the MS is also designed such that all the nominated SPs are visited with optimal tour path.

3 System model

3.1 Network deployment model

This article considers a WSN made up of n number of sensor nodes and an MS. The MS travels across the target area and sojourns at specific sites to gather the sensory data. To illustrate the efficacy of the proposed protocol, three different types of deployment scenarios are considered in this article. Each node deployment follows a probability distribution model; correspondingly, they fulfil the requirement of a specific application. Generally, the uniform and non-uniform distributions of sensor node deployments are implemented in WSN. Here, uniform deployments allow sensor nodes to be deployed in such a way that node density is uniformly distributed across the network; while non-uniform deployment (Huang and Savkin 2017) implies that sensor nodes should be sparsely populated in some regions and densely populated in some other regions. Uniformly distributed WSN is most feasible for area coverage (Harizan and Kuila 2020; Khedr et al. 2018) monitoring which demands continuous surveillance of each part of the target area. This kind of monitoring is highly desirable in battlefield surveillance, where malicious intruders are likely to be detected immediately. The non-uniform deployment is suitable for scenarios where the demand for sensing coverage degree is variable. Several military applications require different degrees of detection ability at different points in the target area and therefore, do not have a stringent restriction on detection timing. Here, the non-uniform deployment will be more suitable. Moreover, at times to safeguard a central camp containing weaponry from the enemies a larger number of sensors are necessarily deployed around it, while other non-critical objects are sparsely surrounded by the sensors. This produces none other than Gaussian-distributed deployment (Wang et al. 2012) which is a special case of non-uniform distribution. Here, sensor nodes are randomly but more densely deployed around a central object, and further the location from the centre lower the likelihood of the deployment density. Hence, this article examines the proposed methodology by employing 2D uniform, Gaussian, and non-uniformly distributed deployments in order to meet the criteria of immediate intrusion detection and detection with higher probability at crucial points, respectively. It should be noted that deployments following the above-stated distributions obey their corresponding bi-variate probability density function (pdf) $f(x, y)$ whose mathematical notations are given below:

- A $2D$ uniform distribution in the range $[a, b]$ has a pdf as:

$$f(x,y) = \begin{cases} \frac{1}{b-a} \; for & a \le x, y \le b \\ 0 \quad for & x, y < a \;\; or \;\; x, y > b \end{cases} \tag{1}$$

  where $a = 0$ and $b = 200$ are taken for the experimental use.
- A $2D$ Gaussian distribution having similar mean $(\mu_x = \mu_y)$ and similar standard deviation $(\sigma_x = \sigma_y)$ in both the dimensions (X and Y) has a pdf as:

$$f(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu)^2+(y-\mu)^2}{2\sigma^2}}$$

However, a major issue caught in deployment with traditional Gaussian distribution refers to the unbounded deployment of sensors when the standard deviation value rises above some threshold. Therefore, to acquire a confined deployment within the boundary of the target area truncated Gaussian distribution (Wang et al. 2012) is employed in this paper whose pdf is given as:

$$f^T(x,y,\sigma) = \frac{f(x,y,\sigma)}{\int_{-\frac{L}{2}}^{\frac{L}{2}} \int_{-\frac{L}{2}}^{\frac{L}{2}} f(x,y,\sigma) dydx}. \tag{2}$$

The above-stated node distributions are graphically presented in Fig. 1a–i exhibit the (truncated) Gaussian, non-uniform, and uniformly distributed deployments, respectively for variable number of nodes (n). The influence of varying node densities on WSN performance metrics is elaborated in Sect. 6.1. For the experimental purpose, the $2D$ truncated Gaussian distribution parameters are assumed as $(\mu_x = \mu_y = 100)$, $(\sigma_x = \sigma_y = 40)$ in both the dimensions X and Y of the target area (where $0 \le X \le 200, 0 \le Y \le 200$).

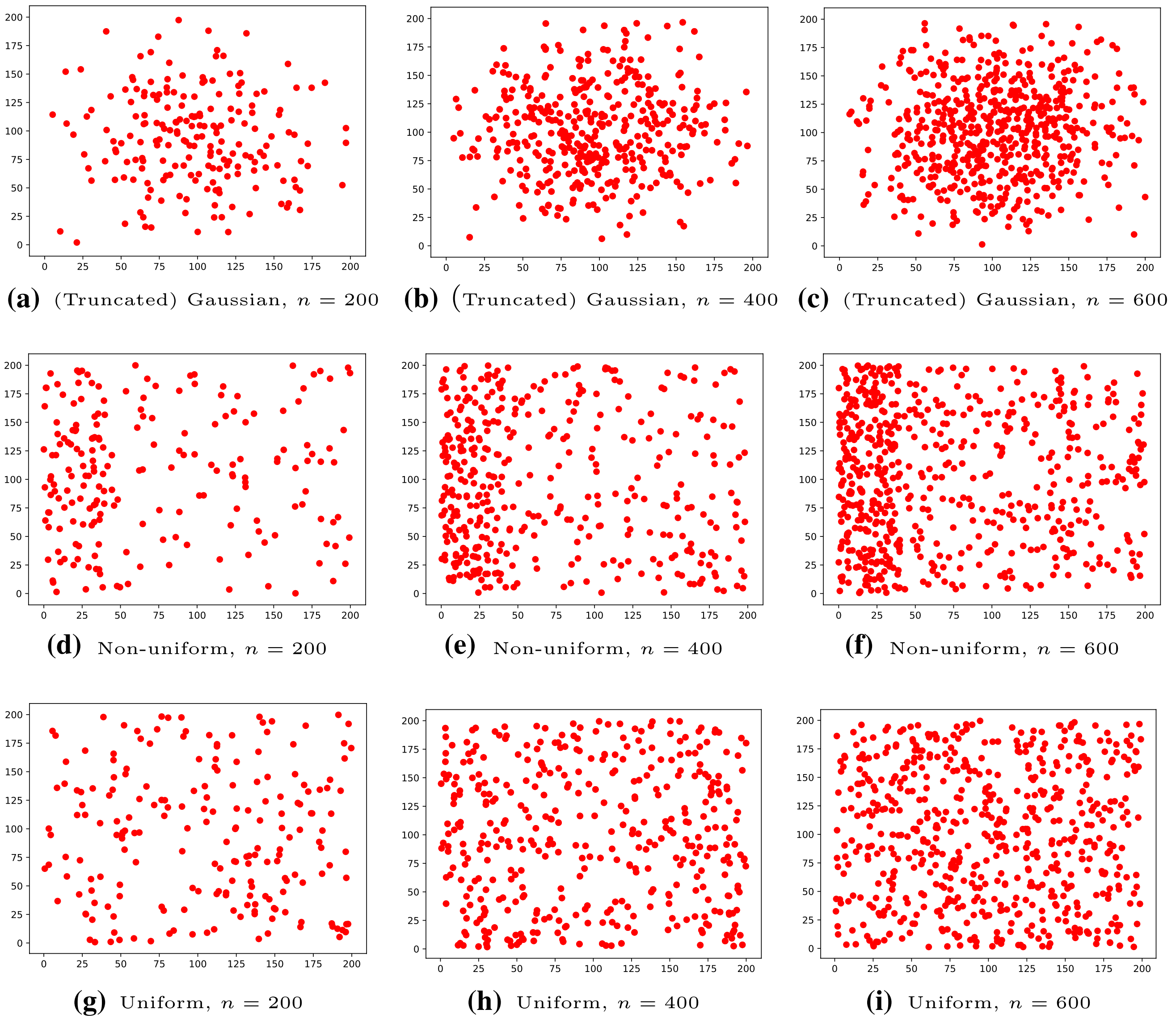
3.2 Rudimentary assumptions

This part comprises the rudimentary assumptions used throughout the paper as follows:

- All deployed sensor nodes are usually static throughout their lifetime. However, they may be moved under external interferences, and the movement of a sensor node is followed by the random walk mobility model (Qin et al. 2018).
- The geographical location of every sensor node is known by means of a global positioning system or some localization algorithms (Ukani et al. 2019; Wang et al. 2018).
- Every sensor node is identical with respect to their battery power, sensing range, and communication range.
- Each sensor node can regulate its transmission power level based on the propagation distance by means of some power control strategies (Gajjar et al. 2016; Lai et al. 2012).
- The sensor nodes are well synchronized in respect of their timer values (Benzaïd et al. 2017; Gherbi et al. 2016).
- According to the references (Harizan and Kuila 2020; Shivalingegowda and Jayasree 2020) to ensure connectivity, inter-cluster communication range (R_{com}) should be at least twice the size of intra-cluster communication range (R_C) i.e. $R_{com} \ge 2 * R_C$.

3.3 Relevant terminologies

The fundamental terminologies used in throughout this paper are listed below.



(a) (Truncated) Gaussian, $n = 200$ **(b)** (Truncated) Gaussian, $n = 400$ **(c)** (Truncated) Gaussian, $n = 600$

(d) Non-uniform, $n = 200$ **(e)** Non-uniform, $n = 400$ **(f)** Non-uniform, $n = 600$

(g) Uniform, $n = 200$ **(h)** Uniform, $n = 400$ **(i)** Uniform, $n = 600$

Fig. 1 Different deployment distributions of WSNs with varying node densities

(i) Sets $\mathcal{S}$ and $\mathcal{S}_{CH}$: Set $\mathcal{S}$ denotes the collective representation of n number of distinct sensor nodes deployed over the target area of size $A \times A\ unit^2$. After accomplishing clustering over $\mathcal{S}$, a new set $\mathcal{S}_{CH}$ composed of $\mathcal{N}_{CH}$ number of CHs is generated which satisfies the following set inclusion $\mathcal{S}_{CH} \subset \mathcal{S}$

(ii) CH density function(F_{den}): It measures the population density of a CH in terms of number of CHs present in its ($2 * R_{com}$) range. For each CH s_i, it can be computed by a bijective function $F_{den} : \mathcal{S}_{CH} \rightarrow \mathcal{S}_{den}$ as:

$$F_{den}(i) = \frac{no.\ of\ other\ CHs\ in\ (2 * R_{com})\ of\ i^{th}\ CH + 1}{\mathcal{N}_{CH}} \quad (3)$$

where $\mathcal{N}_{CH}$ denotes the total number of CHs in the network and $0 \le F_{den}(i) < 1$. The } + 1’ in the numerator part indicates the inclusion of the i^{th} CH itself along with other CHs in its $2 * R_{com}$ range. Set $\mathcal{S}_{den}$ holds all such F_{den} values of the corresponding CHs.

(iii) Highest density node (hdn): It denotes the CH s_j possessing highest CH Density value among all the CHs in $\mathcal{S}_{CH}$ i.e.,

$$\underset{j \in \mathcal{S}_{CH}}{argmax}\, F_{den}(j) := \{j \mid \forall i \in \mathcal{S}_{CH} : F_{den}(i) < F_{den}(j)\}$$

where arg max returns a singleton consisting the id of the highest density CH. Hence, we can write

$$hdn = \arg\max_{j \in \mathcal{S}_{CH}} F_{den}(j) \tag{4}$$

(iv) 2-Circle intersection $(2 - C\ inter)$: It should be noted that communication range (R_{com}) of each sensor forms a circular shape in $2 - D$ space. Moreover, the theory of circle-circle intersection tells that each pair of circle can be intersected at one or two intersection point(s). A **2-Circle intersection** $(2 - C\ inter)$ refers only such circle intersection having two intersection points which is pictorially presented in Fig. 2. In this figure, the red dots denote the CHs while the black dots represent the 2-Circle intersection points. Consequently, for each CH s_i, a **2-C set** can be constructed which contains the ids of other CHs with whom s_i has a $2 - C\ inter$ i.e.,

$$S_i^{2-C} = \{j \mid j \in \mathcal{S}_{CH} \wedge dist(i,j) < 2 * R_{com}\} \tag{5}$$

From Fig. 2, the 2-C set of the following CHs are clearly seen as: $\mathbf{S_1^{2-C}} = \{2,4\}$, $\mathbf{S_2^{2-C}} = \{1,3,4\}$, $\mathbf{S_3^{2-C}} = \{2,4,5\}$, $\mathbf{S_4^{2-C}} = \{1,2,3,5\}$, and $\mathbf{S_5^{2-C}} = \{3,4\}$. In addition, the respective $Inter_i^{2-C}$ set is preserved for each CH, whose each element is a tuple of the intersection points with the CH and other respective CHs in set S_i^{2-C} i.e.,

$$Inter_i^{2-C} = \{((x_1^{ik}, y_1^{ik}), (x_2^{ik}, y_2^{ik})) \mid k \in S_i^{2-C}\}. \tag{6}$$

Figure 2 shows $Inter_2^{2-C} = \{((x_1^{21}, y_1^{21}), (x_2^{21}, y_2^{21})), ((x_1^{23}, y_1^{23}), (x_2^{23}, y_2^{23})), ((x_1^{24}, y_1^{24}), (x_2^{24}, y_2^{24}))\}$ which is a collection of three tuples each of which represents two intersection points generated by the intersection of CH s_2 with CHs $s_1, s_3, s_4 \in S_2^{2-C}$.

(v) Common overlapping region (COR): For p number of intersected circles (or CHs), there can be p or less than p number of common intersection points (CIP) and the enclosure of these points results a region called common overlapping region (COR). The set of CIPs is mathematically denoted as:

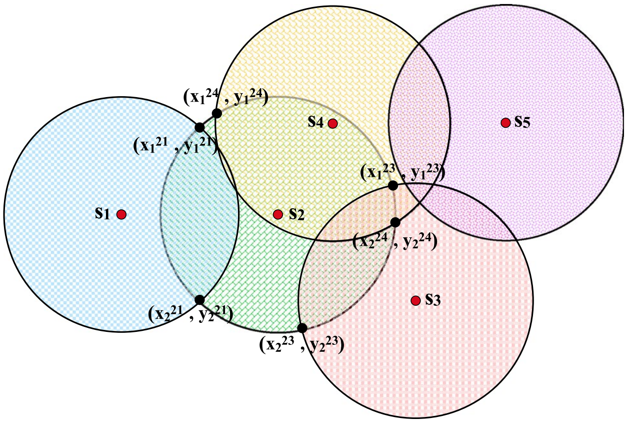


Fig. 2 $2 - C$ intersection for CH s_2

$$S_{cip}^{p} = \{(x_i, y_i) \in \mathbb{R} \times \mathbb{R} \mid 0 \le x_i, y_i \le 200 \wedge i = 1, 2, \dots, q \quad where \quad q \le p\} \tag{7}$$

Figure 3a–c shows the CORs constructed from 2, 3 and 4 circle intersections respectively. According to Eq. 7 , in Fig. 3c, $S_{pts}^4 = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$. It is obvious that placing a Mobile Sink (MS) within any COR ensures a simultaneous one-hop data gathering from p number of CHs. A collection of such p number of CHs having COR constructs a common overlapping set (**COS**) which is mathematically defined as:

$$COS^p = \Big\{j \mid j \in \boldsymbol{\mathcal{S}_{CH}} \quad and \quad \bigcap R_{com}(j) \neq \emptyset\Big\} \quad where \quad |COS^p| = p \tag{8}$$

(vi) Maximum length COS: In order to attain one-hop data gathering from maximum possible number of CHs, the proposed protocol quests after maximum length COS. This leads to selecting COR having maximum number of CH intersection among all the CORs. The following Fig. 4 shows the correct placement of MS with a red tick as it refers to a COR of highest number of CHs (here 4).

3.4 Problem formulation

In this study, our primary objective is to discover an optimal set of MS sojourn points $\mathcal{S}_{soj}$ that guarantees a compromise between energy preservation and data delivery delay. This can be obtained by minimizing the number of such SPs $(\mathcal{N}_{sp})$ while securing a one-hop connectivity to the CHs. Hence, the minimization problem of MS sojourn points can be formulated as:

$$\mathbf{P1}: \quad \text{minimize} \quad \boldsymbol{\mathcal{N}_{sp}} \tag{9a}$$

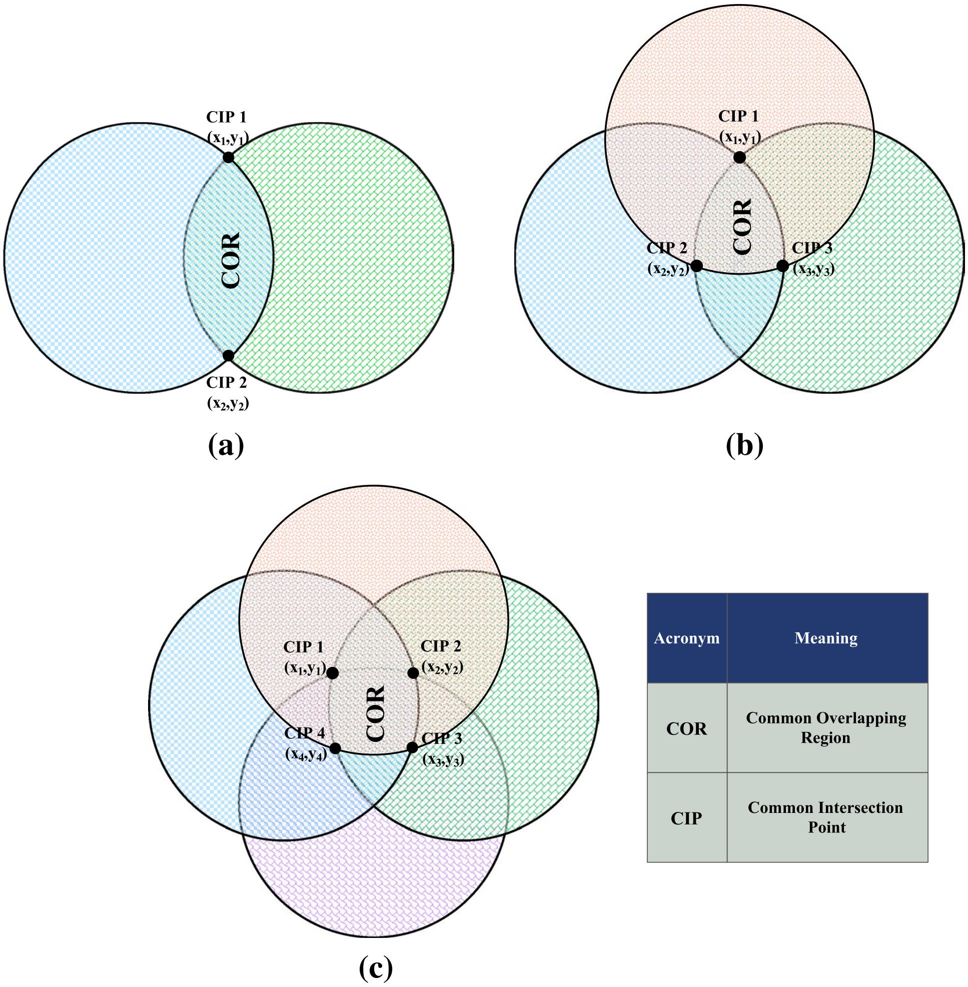
$$\text{subject to} \quad \sum_{i=1}^{\mathcal{N}_{sp}} \mathcal{X}_{ij} \ge 1, \qquad \forall j \in \mathcal{S}_{CH} \tag{9b}$$

where constraint 9b states that for each cluster head $j \in \mathcal{S}_{CH}$ there exists atleast one SP to obtain one-hop data gathering. $\mathcal{X}_{ij}$ is a binary variable which is defined by:

$$\mathcal{X}_{ij} = \begin{cases} 1 & dist(i,j) \le \delta_{th} \\ 0 & \text{otherwise} \end{cases}$$

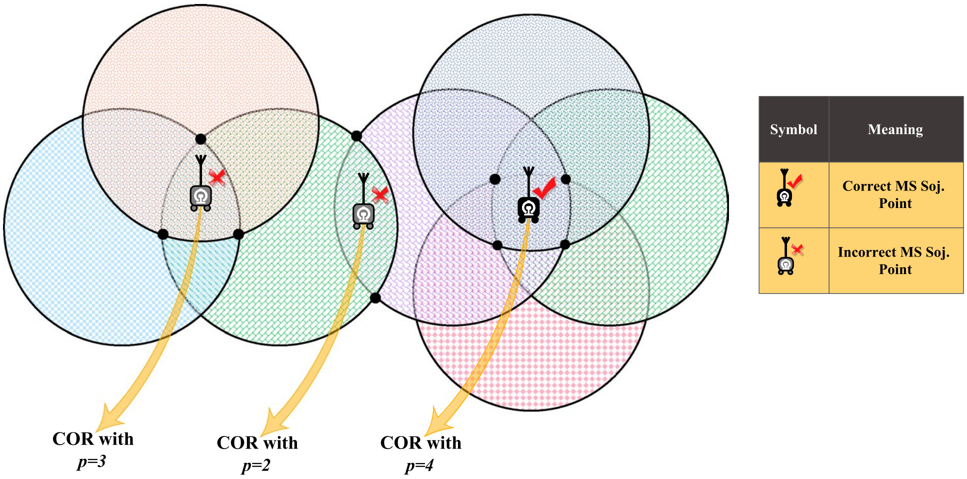
where $dist(i,j)$ is the euclidean distance between CH $j \in \mathcal{S}_{CH}$ and sojourn point $i \in \mathcal{S}_{soj}$ and δ_{th} is the threshold distance for

Fig. 3 Common overlaps



Acronym	Meaning
COR	Common Overlapping Region
CIP	Common Intersection Point

Fig. 4 Maximum COR finding



Symbol	Meaning
✓	Correct MS Soj. Point
✗	Incorrect MS Soj. Point

free-space energy model (as defined in Sect. 6.2). The latter part of the proposed protocol constructs an optimal path for the MS visiting every SP exactly once and returning back to the starting SP. This kind of optimal path mainly aims to reduce the total travelling distance of the MS tour which in turn minimizes the data gathering delay. This refers to nothing but a $\mathcal{N}_{sp}$ city travelling salesman problem and according to Miller–Tucker–Zemlin the MS tour minimization problem can be formulated as:

$$\mathbf{P2}: \quad \text{minimize} \quad \sum_{i=1}^{\mathcal{N}_{sp}} \sum_{j \neq i, j=1}^{\mathcal{N}_{sp}} \mathcal{C}_{ij}\mathcal{X}_{ij} \tag{10a}$$

$$\text{subject to} \quad \sum_{i=1, i \neq j}^{\mathcal{N}_{sp}} \mathcal{X}_{ij} = 1, \quad \forall j = 1, \ldots, \mathcal{N}_{sp} \tag{10b}$$

$$\sum_{j=1, j \neq i}^{\mathcal{N}_{sp}} \mathcal{X}_{ij} = 1, \quad \forall i = 1, \ldots, \mathcal{N}_{sp} \tag{10c}$$

$$\mathcal{U}_i - \mathcal{U}_j + \mathcal{N}_{sp}\mathcal{X}_{ij} \le \mathcal{N}_{sp} - 1 \qquad 2 \le i \neq j \le \mathcal{N}_{sp} \tag{10d}$$

$$\mathcal{U}_i \ge 0 \quad \forall i = 1, \ldots, \mathcal{N}_{sp} \tag{10e}$$

$$\mathcal{X}_{ij} \in \{0, 1\} \quad \forall i, j = 1, \ldots, \mathcal{N}_{sp} \tag{10f}$$

where $\mathcal{C}_{ij}$ is the travelling distance from SP i to SP j and $\mathcal{X}_{ij}$ is a binary variable that is defined by:

$$\mathcal{X}_{ij} = \begin{cases} 1 & \text{if SP } j \text{ is reached from SP } i \\ 0 & \text{otherwise} \end{cases}$$

Constraints 10b and 10c refer to exactly one entry for a SP $j \in \mathcal{S}_{soj}$ and exactly one exit for a SP $i \in \mathcal{S}_{soj}$, respectively. Eq. 10d denotes the constraint for subtour elimination where $\mathcal{U}_i$ is the auxiliary variable $\forall i = 1, \ldots, \mathcal{N}_{sp}$.

4 Proposed work

The proposed protocol comprises the following phases:

4.1 Phase 1: set-up phase

All the sensor nodes in this phase participate in the clustering process by adopting any standard clustering algorithm like (Yu et al. 2012; Sabet and Naji 2015) to evenly partition the whole network into a number of clusters. Based on the CH information (ID, location, etc.), BS nominates a set of optimal MS Sojourn points (SP) exploiting which a closed path is built where each SP will be visited only once by the MS for data gathering.

4.2 Phase 2: routing phase

The working principle of the routing phase can be categorized into two sub-phases as (i) optimal sojourn points nomination and (ii) MS traversal path construction.

4.2.1 Optimal sojourn points nomination

In order to achieve better energy efficiency, this sub-phase aims to choose a set of MS sojourn points which ensure a one-hop data collection. However, inefficient one-hop data collection may increase the number of SPs, which is not feasible for delay-sensitive networks. Hence appropriate selection of SPs plays a vital role in MS based data gathering. The proposed protocol nominates the SPs in such a way that each SP is surrounded by the maximum possible number of cluster heads (CH) in the one-hop distance. It helps to attain a well regulation of the trade-off between energy consumption and data delivery latency. Figure 5 presents an algorithm of the optimal SP nomination in the form of a concise flow diagram. This flow diagram is an iterative process whose every iteration leads to an SP. The iterative process continues till all the CHs are covered in the one-hop distance by the nominated SPs. An outline of the proposed flow diagram is given below:

- **Input:**
  - Set $\mathcal{S}_{soj}$ will hold the Sojourn points (SP) found in each iteration. Initially, $\mathcal{S}_{soj}$ is null.
  - In order to safeguard from data loss $\mathcal{S}_{CH}$ remains intact and its replica ($\overline{\mathcal{S}_{CH}}$) is employed.
- **Process 1:** Calculates the CH density ($F_{den}()$) value for every CH $\in \overline{\mathcal{S}_{CH}}$ using Eq. 3
- **Process 2:**
  - Determines highest density node (hdn) among all CHs $\in \overline{\mathcal{S}_{CH}}$ using Eq. 4.
  - Calculate 2-Circle intersection set S_{hdn}^{2-C} using Eq. 5.
- **Process 3:** Calculates maximum length common overlapping set (COS_{max}^{p}) for hdn by invoking the function $Max\_COS(hdn, S_{hdn}^{2-c})$. In other words, COS_{max}^{p} refers to the set of CHs covered by the forthcoming SP in the current iteration.
- **Process 4:**
  - Calculates the corresponding set of common intersecting points (S_{cip}^{p}) by invoking the function $INTER\_PTS(COS_{max}^{p})$.
  - Finds the coordinates $\{(x_{cen}, y_{cen})\}$ of j^{th} SP by calculating the centroid of all CIPs.
- **Process 5:** After every iteration, value updating is performed as:

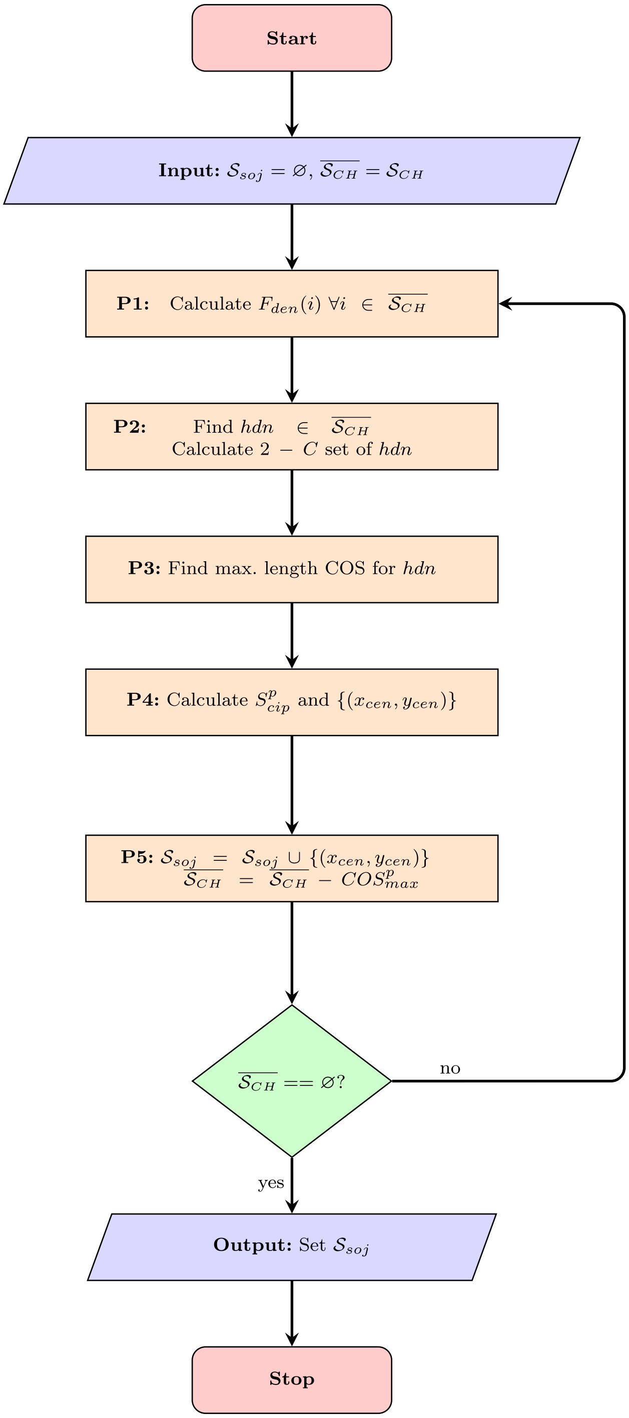

Start
Input: $\mathcal{S}_{soj} = \varnothing$, $\overline{\mathcal{S}_{CH}} = \mathcal{S}_{CH}$
P1: Calculate $F_{den}(i)$ $\forall i \in \overline{\mathcal{S}_{CH}}$
P2: Find $hdn \in \overline{\mathcal{S}_{CH}}$
Calculate $2-C$ set of hdn
P3: Find max. length COS for hdn
P4: Calculate S^{p}_{cip} and $\{(x_{cen}, y_{cen})\}$
P5: $\mathcal{S}_{soj} = \mathcal{S}_{soj} \cup \{(x_{cen}, y_{cen})\}$
$\overline{\mathcal{S}_{CH}} = \overline{\mathcal{S}_{CH}} - COS^{p}_{max}$
$\overline{\mathcal{S}_{CH}} == \varnothing$?
no
yes
Output: Set $\mathcal{S}_{soj}$
Stop

Fig. 5 Flow diagram of the proposed SP nomination approach

- Updates $\mathcal{S}_{soj}$ by adding the coordinates of the SP found in the current iteration.
- Updates $\overline{\mathcal{S}_{CH}}$ by eliminating the set of covered CHs (COS^{p}_{max}) from it (as the covered CHs will not be considered in the next iteration).

- **Decision:** Subsequently, decision making is also performed to check whether all CHs are covered or not as:
  - **IF** ($\overline{\mathcal{S}_{CH}} == \varnothing$) i.e. all CHs are covered **then** Go to **Output**
  - **ELSE** Go to **Process 1** i.e.start of the next iteration.

- **Output:** A set of sojourn points $\mathcal{S}_{soj} = \left\{ (x^{soj}_{j}, y^{soj}_{j}) \mid j = 1, ..., \mathcal{N}_{sp} \wedge \left(\mathcal{N}_{sp} \ll \mathcal{N}_{CH} \right) \right\}$

It is to be noted that the aforementioned algorithm (Fig. 5) will not run in every round in order to mitigate the computational complexity. It will be invoked if and only if re-clustering takes place. Upon re-clustering, it generates a new set of SPs to maintain the adaptability of the network. For a better understanding of the readers, a synthetic example of the proposed algorithm is illustrated below.

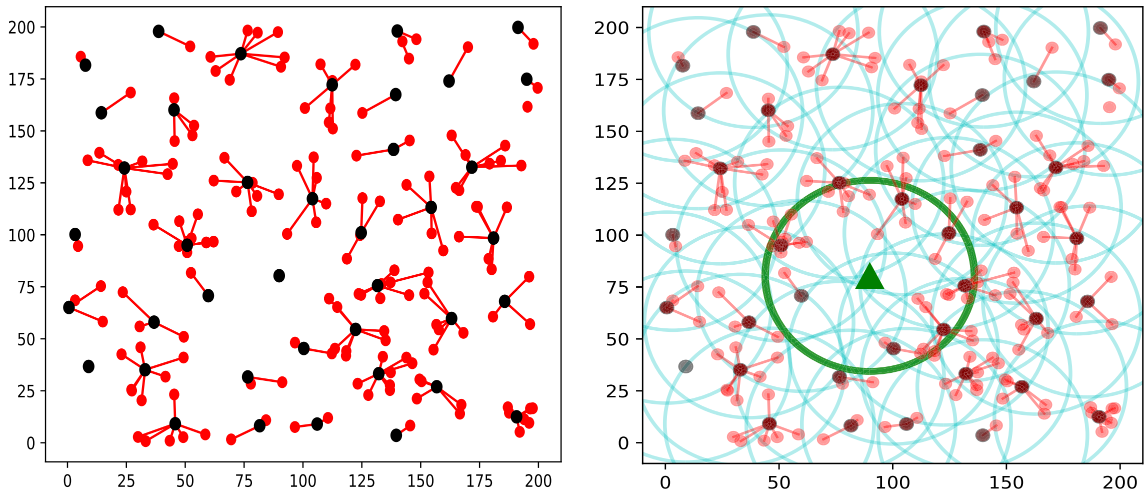
4.2.2 Illustration of SP nomination

Consider a WSN of 200 sensor nodes i.e., $\mathbf{S} = \left\{ s_0, s_1, \ldots, s_{199} \right\}$ which are deployed over a target area of size $200 \times 200\, unit^2$. The node deployment follows the uniform distribution model (Eq. 1). After clustering, from the set $\mathbf{S}$, the following set of cluster heads $\boldsymbol{\mathcal{S}}_{\mathbf{CH}}$ is generated as:

$$\begin{aligned}\boldsymbol{\mathcal{S}}_{\mathbf{CH}} = [&13, 46, 119, 14, 2, 186, 71, 92, 56, 62, 38, 129,\\ &174, 91, 28, 132, 165, 115, 53, 35, 16,\\ &197, 87, 149, 11, 193, 156, 89, 138,\\ &81, 116, 93, 154, 58, 180, 94, 29, 185, 68, 163]\end{aligned}$$

For the sake of simplicity, only the subscript i of each CH s_i is used to denote its corresponding ID. In Fig. 6a, the black dots refer to the Cluster Heads (CH) and their corresponding member nodes are shown in red dots. The association between a member and its respective CH are shown by the red lines. Let's have a close look at three successive iterations.

INPUT: $\mathcal{S}_{soj} = \varnothing, \overline{\mathcal{S}_{CH}} = \mathcal{S}_{CH}$



(a) Clustering on set **S** **(b)** hdn and its R_{com}

Fig. 6 Set up phase and Highest density node (hdn) finding

Iteration 1

P1: Calculates $F_{den}()$ value of each $CH \in \overline{\mathcal{S}_{CH}}$ using Eq. 3

P2:
- Finds **hdn** using Eq. 4. Here, it is CH having ID **163** where $F_{den}(163)$ is 0.65
- Calculates $\mathbf{S^{2-C}_{163}} = \{\mathit{11}, \mathit{16}, \mathit{29}, \mathit{38}, \mathit{53}, \mathit{56}, \mathit{58}, \mathit{68}, \mathit{71}, \mathit{81}, \mathit{89}, \mathit{93}, \mathit{94}, \mathit{115}, \mathit{116}, \mathit{129}, \mathit{132}, \mathit{138}, \mathit{149}, \mathit{154}, \mathit{156}, \mathit{174}, \mathit{180}, \mathit{185}, \mathit{193}\}$ using Eq. 5

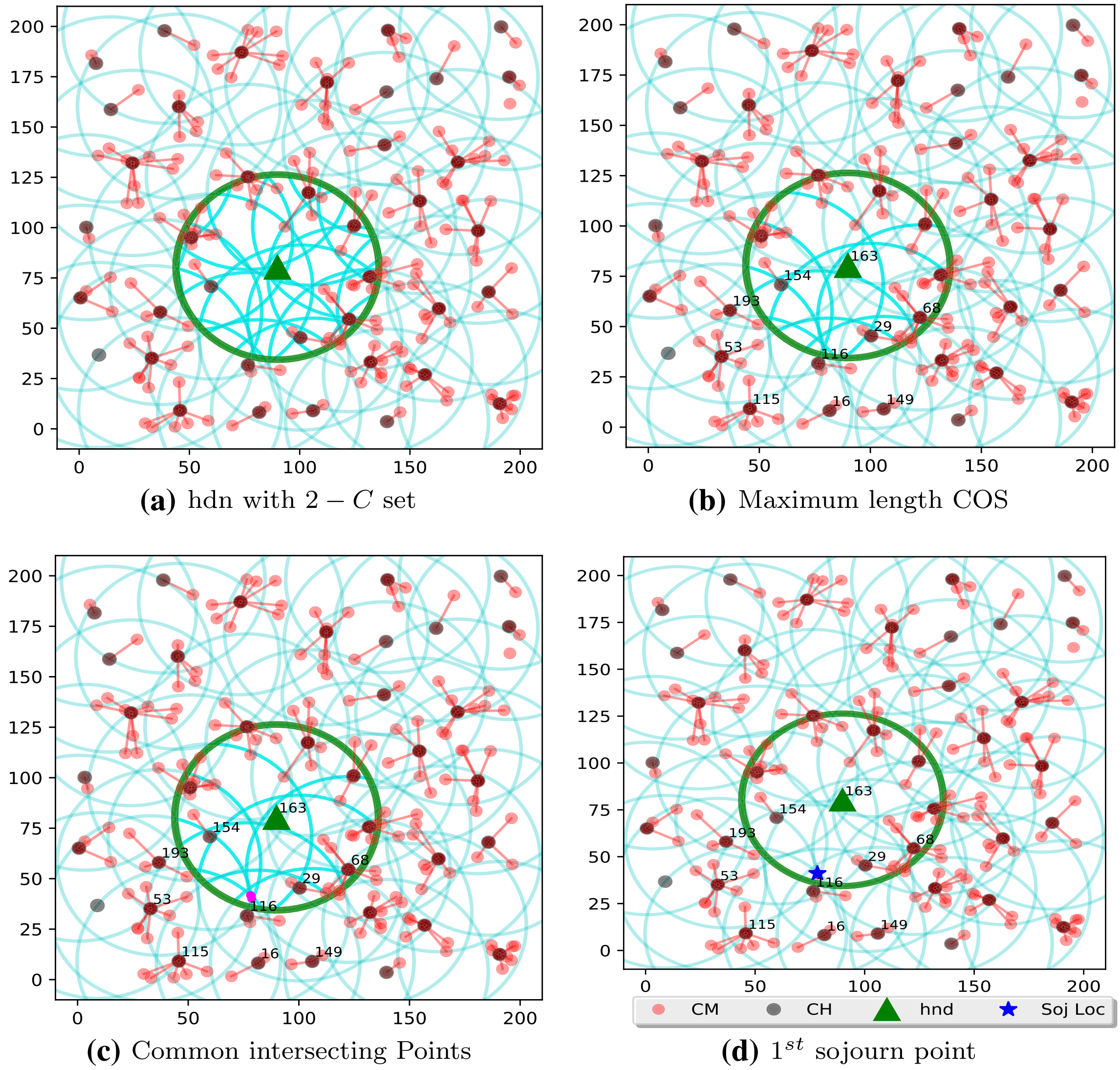
P3: Calculates $\mathbf{COS^{p}_{max}}$=$\{\mathit{16}, \mathit{29}, \mathit{53}, \mathit{68}, \mathit{115}, \mathit{116}, \mathit{149}, \mathit{154}, \mathit{163}, \mathit{193}\}$ for $hdn =$ 163 by invoking function $MAX\_COS(hdn, S^{2-C}_{hdn})$

P4:
- Calculates $\mathbf{S^{p}_{cip}}$= $[\{x : 78.67,\ y : 40.14\}, \{x : 78.50,\ y : 41.47\}, \{x : 78.15,\ y : 41.82\}]$ by invoking function $INTER\_PTS(COS^{p}_{max})$
- Calculates $\mathbf{x_{cen}} = 78.447$ and $\mathbf{y_{cen}}$=41.146 i.e., $(\mathbf{x^{soj}_{1}}, \mathbf{y^{soj}_{1}}) = (\mathbf{x_{cen}}, \mathbf{y_{cen}})$

P5: Value Updating:
- $\mathcal{S}_{\mathbf{soj}} = \{(\mathbf{x^{soj}_{1}}, \mathbf{y^{soj}_{1}})\}$
- $\overline{\mathcal{S}_{CH}} = \overline{\mathcal{S}_{CH}} - \mathbf{COS^{p}_{max}} = \{\mathit{56}, \mathit{71}, \mathit{174}, \mathit{46}, \mathit{119}, \mathit{13}, \mathit{186}, \mathit{14}, \mathit{129}, \mathit{11}, \mathit{2}, \mathit{38}, \mathit{81}, \mathit{132}, \mathit{89}, \mathit{92}, \mathit{165}, \mathit{62}, \mathit{91}, \mathit{156}, \mathit{28}, \mathit{35}, \mathit{94}, \mathit{87}, \mathit{197}, \mathit{58}, \mathit{180}, \mathit{93}, \mathit{138}, \mathit{185}\}$

D: Since, $\overline{\mathcal{S}_{CH}} \neq \varnothing$, Continue for another iteration (Iteration 2)

In Fig. 6b, highest density node $hdn = 163$ and its communication radius R_{com} are marked by green triangle and green circle respectively. The R_{com} of each CH except the hdn is marked by a light cyan circle. Furthermore, the non-CH nodes are blurred as we are only concerned about the CHs. As shown in Fig. 7a, the dark cyan chords inside the dark green circle presents the $2-C$ *inter* for $hdn = 163$. Figure 7b exhibits the maximum length COS (COS^{p}_{max}) for $hdn = 163$ with a lesser number of dark cyan chords than Fig. 7a. Figure 7b shows the corresponding set of CIPs (S^{p}_{cip}) by magenta dots and Fig. 7d depicts the first SP by blue coloured star. Figure 8a, b exhibits the scenarios before and after updating of $\overline{\boldsymbol{\mathcal{S}}_{\mathbf{CH}}}$, respectively. The same process is repeated in Iteration 2 and 3 (as shown in Figs. 9a, 10, 11c) over updated set $\overline{\mathcal{S}_{CH}}$ set each iteration concludes with one optimal SP (in Figs. 9b and 11a). Iterations will continue until all the CHs are covered by their respective SPs i.e. $\overline{\boldsymbol{\mathcal{S}}_{\mathbf{CH}}} = \varnothing$. The presented synthetic example lasts for 6 iterations and hence generates 6 MS sojourn points (in Fig. 11d) which assures a one-hop data collection from all the 40 CHs.



(a) hdn with $2 - C$ set

(b) Maximum length COS

(c) Common intersecting Points

(d) 1^{st} sojourn point

Fig. 7 Iteration 1 of the proposed algorithm (Fig 5)

Iteration 2

P1: Calculates $F_{den}()$ of each $CH \in \overline{\mathcal{S}_{CH}}$

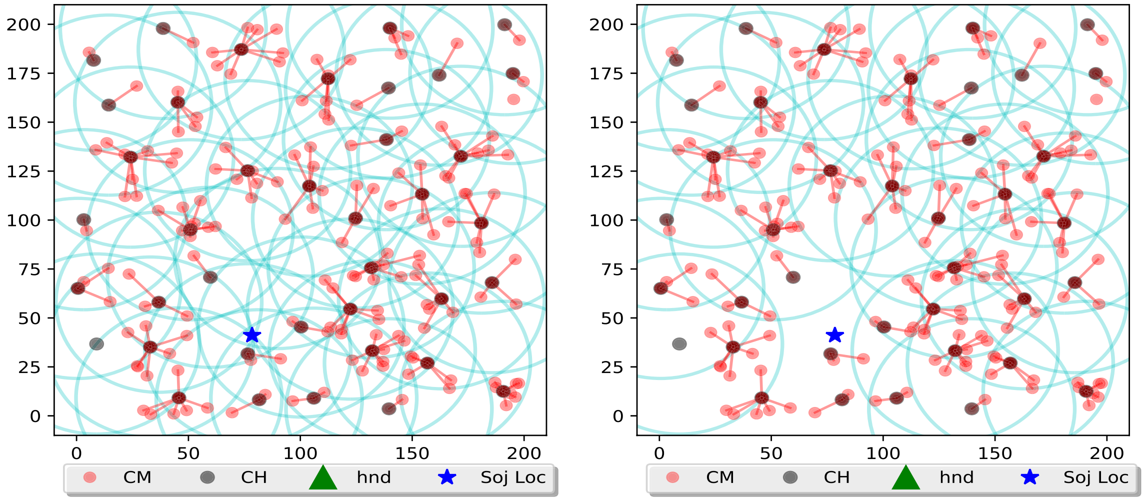
P2:
- Finds $\mathbf{hdn} = \mathbf{185}$ where $F_{den}(185)$ is 0.45
- Calculates $\mathbf{S^{2-C}_{185}} = \{28, 35, 38, 58, 62, 81, 87, 89, 91, 92, 93, 94, 132, 138, 156, 180, 197\}$

P3: Calculates $\mathbf{COS^{p}_{max}}$=$\{58, 87, 93, 94, 138, 156, 185, 197\}$ for $hdn = 185$

P4:
- Calculates $\mathbf{S^{p}_{cip}} = [\{x : 144.51,\ y : 95.49\}, \{x : 144.49,\ y : 95.49\}, \{x : 147.80,\ y : 103.17\}, \{x : 139.80,\ y : 99.45\}, \{x : 144.50,\ y : 95.49\}]$
- Calculates $\mathbf{x_{cen}}$ =144.22 and $\mathbf{y_{cen}}$ =97.818 i.e., $(\mathbf{x^{soj}_{2}}, \mathbf{y^{soj}_{2}}) = (\mathbf{x_{cen}}, \mathbf{y_{cen}})$

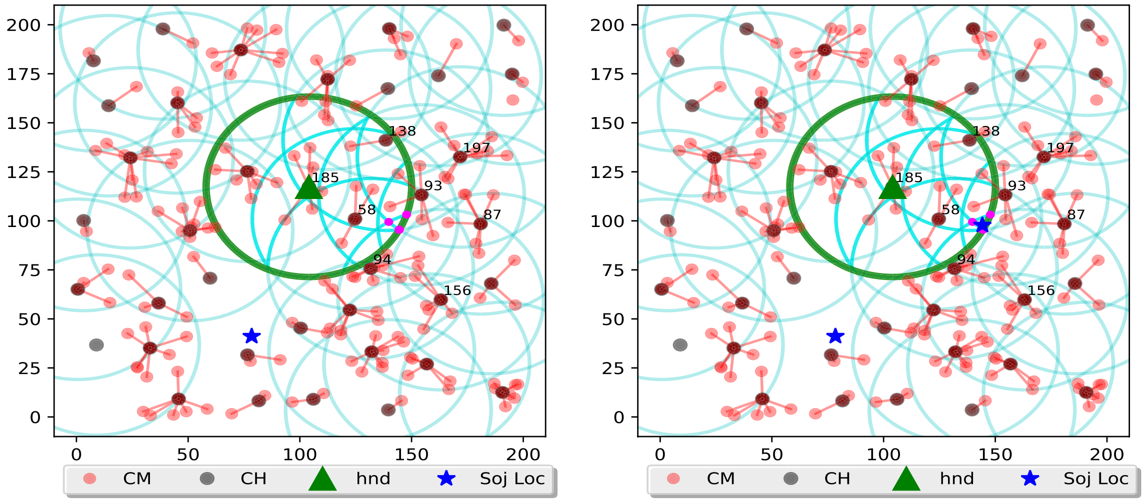
P5: Value Updating:
- $\mathcal{S}_{\mathbf{soj}} = \{(\mathbf{x^{soj}_{1}}, \mathbf{y^{soj}_{1}}), (\mathbf{x^{soj}_{2}}, \mathbf{y^{soj}_{2}})\}$
- $\overline{\mathcal{S}_{CH}} = \overline{\mathcal{S}_{CH}} - \mathbf{COS^{p}_{max}} = \{56, 11, 71, 89, 165, 174, 186, 119, 2, 62, 92, 46, 81, 13, 28, 129, 14, 38, 35, 132, 180, 91\}$

D: Since, $\overline{\mathcal{S}_{CH}} \neq \varnothing$, Continue for another iteration (Iteration 3)



(a) Before updating $\overline{\mathcal{S}_{CH}}$ **(b)** After updating $\overline{\mathcal{S}_{CH}}$

Fig. 8 Iteration 1 (continued)



(a) hdn, Max length COS and CIPs **(b)** 2^{nd} sojourn point

Fig. 9 Iteration 2 of the proposed algorithm (Fig 5)

Iteration 3

P1: Calculates $F_{den}()$ of each $CH \in \overline{\mathcal{S}_{CH}}$

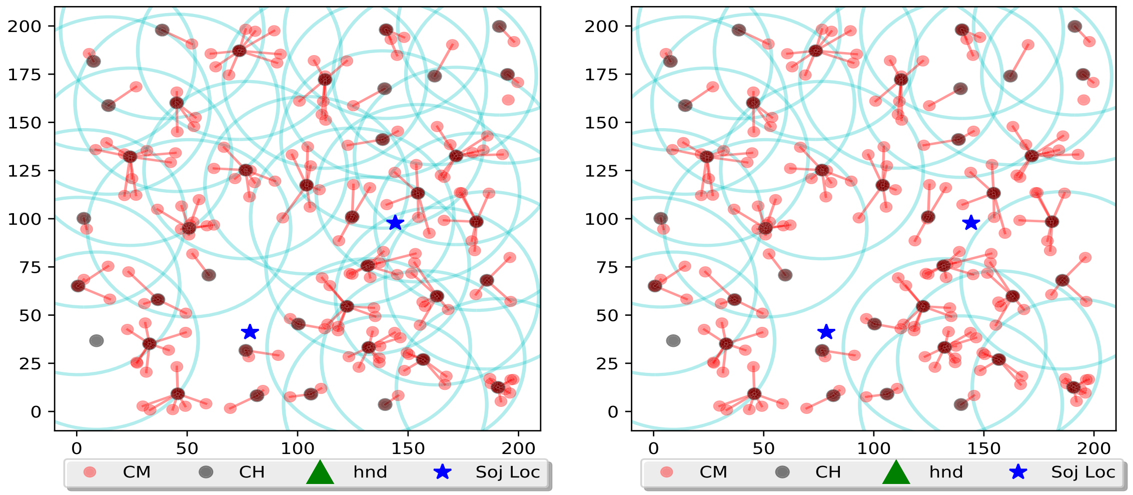
P2:
- Finds $\mathbf{hdn} = \mathbf{91}$ where $F_{den}(91)$ is 0.275
- Calculates $\mathbf{S_{91}^{2-C}}$=$\{\mathit{13}, \mathit{14}, \mathit{28}, \mathit{35}, \mathit{38}, \mathit{46}, \mathit{62}, \mathit{92}, \mathit{132}, \mathit{180}\}$

P3: Calculates $\mathbf{COS_{max}^{p}}$=$\{\mathit{13}, \mathit{14}, \mathit{38}, \mathit{46}, \mathit{91}, \mathit{132}, \mathit{180}\}$ for $hdn = 91$

P4:
- Calculates $\mathbf{S_{cip}^{p}}$ = $[\{x : 43.29,\ y : 152.55\}, \{x : 49.50,\ y : 162.47\}, \{x : 41.08,\ y : 154.62\}]$
- Calculates $\mathbf{x_{cen}}$ =44.62 and $\mathbf{y_{cen}}$ =156.55 i.e., $(\mathbf{x_3^{soj}}, \mathbf{y_3^{soj}}) = (\mathbf{x_{cen}}, \mathbf{y_{cen}})$

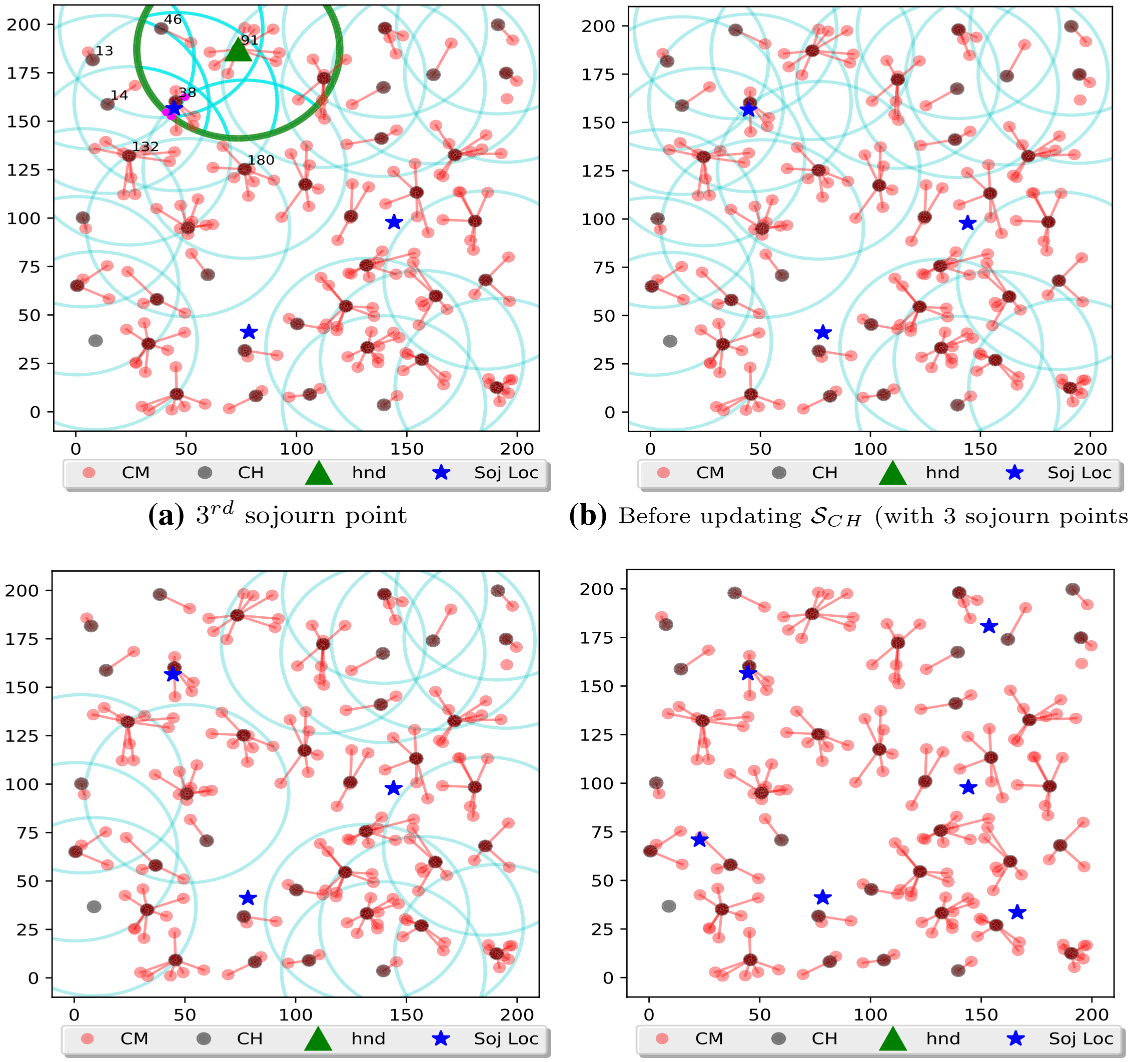
P5: Value Updating:
- $\mathcal{S}_{\mathbf{soj}} = \{(\mathbf{x_1^{soj}}, \mathbf{y_1^{soj}}), (\mathbf{x_2^{soj}}, \mathbf{y_2^{soj}}), (\mathbf{x_3^{soj}}, \mathbf{y_3^{soj}})\}$
- $\overline{\mathcal{S}_{CH}} = \overline{\mathcal{S}_{CH}} - \mathbf{COS_{max}^{p}} = \{\mathit{56}, \mathit{11}, \mathit{71}, \mathit{89}, \mathit{165}, \mathit{174}, \mathit{186}, \mathit{119}, \mathit{2}, \mathit{62}, \mathit{92}, \mathit{81}, \mathit{28}, \mathit{129}, \mathit{35}\}$

D: Since, $\overline{\mathcal{S}_{CH}} \neq \varnothing$, Continue for another iteration (Iteration 4)



(a) Before updating $\mathcal{S}_{CH}$ (with 2 sojourn points) **(b)** After updating $\mathcal{S}_{CH}$ (with 2 sojourn points)

Fig. 10 Iteration 2 (continued)



(a) 3^{rd} sojourn point **(b)** Before updating $\mathcal{S}_{CH}$ (with 3 sojourn points)

(c) After updating $\mathcal{S}_{CH}$ (with 3 sojourn points) **(d)** Iteration 6: all CHs are covered

Fig. 11 Iteration 3 and 6 (final one)

```
Function $MAX\_COS(hdn, S^{2-C}_{hdn})$        // function returning the longest set of cluster head IDs
    $S_1 \leftarrow \varnothing$, $S_2 \leftarrow S^{2-C}_{hdn}$, $S_3 \leftarrow \varnothing$, $COS^p_{max} \leftarrow \varnothing$;
    while $S_2.length > 0$:
        repeat
            $var_1 \leftarrow s_1$ where $s_1 \in S_2$;
            $S_3 \leftarrow S_3 \cup \{var_1\}$;
            for  $k = 2$ to $S_2.length$:
                $var_2 \leftarrow s_k$ where $s_k \in S_2$;
                if $var_2 \in S^{2-C}_{var_1}$ and $S_3.length < 2$:
                    $S_3 \leftarrow S_3 \cup \{var_2\}$;
                else:
                    $count \leftarrow 0$;
                    for $l = 2$ to $S_3.length$:
                        $var_3 \leftarrow s_l$ where $S_l \in S_3$;
                        if $var_2 \in S^{2-C}_{var_3}$:
                            $count = count + 1$
                    if  $count = S_3.length - 1$:
                        $S_3 \leftarrow S_3 \cup \{var_2\}$;
            $S_1 \leftarrow S_1 \cup \{S_3\}$;
            $S_2 \leftarrow S_2 - \{var_1\}$;
        until this end condition;
    // After while loop $\mathbf{S_1} = \{K_1, ..., K_i, K_{i+1}, ..., K_n\}$ where $K_i = \{k_1, .., k_{m_i}\}$ and $1 \le m_i \le m_{i+1}$;
    $COS^p_{max} \leftarrow K_n$ ;
    return $COS^p_{max}$
```

```
Function $INTER\_PTS(COS^p_{max})$     // function returning the tuples of common intersecting points
    $S^p_{cip} \leftarrow \varnothing$ ;
    for $i \leftarrow 1$ to $COS^p_{max}.length - 1$:
        for $j \leftarrow (i + 1)$ to $COS^p_{max}.length$:
            for $k \leftarrow 1$ to $S^{2-C}_i.length$:
                if $j == S^{2-C}_i[k]$:
                    $x_1 \leftarrow Inter^{2-C}_i[k][1][x^{ij}_1]$            // using Eq.6;
                    $y_1 \leftarrow Inter^{2-C}_i[k][1][y^{ij}_1]$;
                    $x_2 \leftarrow Inter^{2-C}_i[k][2][x^{ij}_2]$;
                    $y_2 \leftarrow Inter^{2-C}_i[k][2][y^{ij}_2]$;
                    $pt_1 \leftarrow (x_1, y_1)$;
                    $pt_2 \leftarrow (x_2, y_2)$;
                if $\forall s_l \in COS^p_{max}$  $dist\,(pt_1, s_l) \le 2 * R_{com}$:
                    $S^p_{cip} \leftarrow S_{pts} \cup \{pt_1\}$;
                if $\forall s_l \in COS^p_{max}$  $dist\,(pt_2, s_l) \le 2 * R_{com}$:
                    $S^p_{cip} \leftarrow S_{pts} \cup \{pt_2\}$;
    return $S^p_{cip}$
```

4.2.3 MS traversal path construction

The second step of the routing phase establishes an MS traversal path to visit each SP for data gathering and returns back to the starting point by following some sequence. It is worth mentioning that such sequence is obliged to produce the shortest length tour among all possible tour sequences in order to fulfil the delay-sensitive requirements of WSN. Now, for $\mathcal{N}_{sp}$ number of SPs in the network, there are $\frac{(\mathcal{N}_{sp}-1)!}{2}$ number of possible tours generated from the brute force approach that is extremely large for the large-scale WSNs. This infers that MS path construction incurs a non-polynomial time solution. Hence, it is highly desirable to exploit an optimization strategy like ACO algorithm (Dorigo and Stützle 2019) to build a near-optimal MS tour in a reasonable amount of time. Inspired by the foraging behaviour of the real ants, ACO employs a number of artificial ants which have distinguishing properties as follows:

- They have memory, i.e., they'll not visit the SP that has been already visited.
- They know the distance among the SPs and tend to select the nearest SP if the pheromone level remains same.

- If the distance of two paths are same they'll tend to choose the path which has more pheromone.

The essential parameters used in ACO Algorithm 1 are described below:

- Transition probability($\mathbf{P_{ij}^k(t)}$): It is the probability of how ant k will choose the SP j while sitting at SP i at time t

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{p \in allowed_k}[\tau_{ip}(t)]^{\alpha}[\eta_{ip}]^{\beta}} & if\ j \in allowed_k \\ 0 & otherwise \end{cases} \tag{11}$$

where $allowed_k$ is the set of SP (s) which is (are) not yet visited by ant k, $\tau_{ij}(t)$ is intensity of pheromone trail between SPs i and j at time t, η_{ij} is visibility of SP j from SP i.e., $\eta_{ij} = \frac{i}{d_{ij}}$, and α and β are parameters to regulate the relative influence of trail versus visibility.

- Pheromone updating ($\tau_{\mathbf{ij}}(\mathbf{t} + \boldsymbol{\mathcal{N}}_{\mathbf{sp}})$): Since there are $\mathcal{N}_{sp}$ number of SPs, after $\mathcal{N}_{sp}$ iterations each ant completes a tour. Subsequently, pheromone trail $\tau_{ij}(t + \mathcal{N}_{sp})$ on edge (i,j) at time $t + \mathcal{N}_{sp}$ will be updated as

$$\tau_{ij}(t + \mathcal{N}_{sp}) = \rho.\tau_{ij}(t) + \Delta\tau_{ij} \tag{12}$$

where $\tau_{ij}(t)$ is pheromone trail on edge (i,j) at time t, ρ is evaporation factor which regulates pheromone reduction, and $\Delta\tau_{ij}$ is total change in pheromone trail between time t and $(t + \mathcal{N}_{sp})$ as shown below

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^k \tag{13}$$

where l is the no of ants and $\Delta\tau_{ij}^k$ is quantity per unit length of trail on edge (i,j) by k^{th} ant between time t and $(t + \mathcal{N}_{sp})$

$$\Delta\tau_{ij}^k = \begin{cases} Q/L_k & \text{if ant k travels on edge}(i,j) \text{ between } t \text{ and } (t + \mathcal{N}_{sp}) \\ 0 & otherwise \end{cases}$$

where Q is constant and L_k is tour length by ant k.

Exploiting ACO, Algorithm 1 results a minimal length MS traversal path based on the fully connected graph $G = (\mathcal{S}_{soj}, \mathcal{E}_{soj})$ where $\mathcal{S}_{soj}$ is the set of all SPs obtained by the flow diagram in Fig. 5, and $\mathcal{E}_{soj}$ is the set of connections among these SPs. In Algorithm 1, Line 3 defines two lists to store the shortest tour length and its corresponding cycle, respectively in each iteration. Initially, l number of ants are placed randomly on $\mathcal{N}_{sp}$ number of SPs. Line 6 suggests the assignment of the initial trail value $\tau_{ij}(0)$ for each edge $(i,j) \in E$ to a constant c. This algorithm further maintains a sp_list^k for each ant k, which contains the already visited SPs by k. The movement of each ant k to SP j while sitting at SP i is determined by the probabilistic measure defined in Eq. 11. Once, an SP is visited by ant k; it is inserted to the respective sp_list in order to maintain the set $allowed_k$. Then, the tour length L_k is computed for each ant k when it has visited all the cities. Among them, the shortest tour is found as well as inserted with the corresponding tour length into min_list and min_len respectively. Line 23 and 24 refer the pheromone trail updating on each edge $(i,j) \in E$. The whole process continues until the number of iterations reaches the threshold value or the convergence criteria is reached (Line 26). The final output of Algorithm 1 is simulated in Fig. 12.

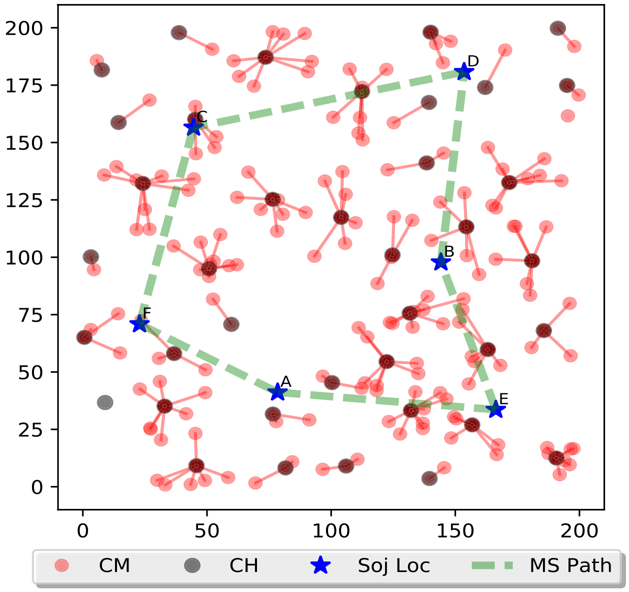


Fig. 12 MS traversal path construction by Algorithm 1

Algorithm 1: MS Path Construction

```
Data: A fully connected graph $G = (\mathcal{S}_{soj}, \mathcal{E}_{soj})$ where $\mathcal{S}_{soj}$ is the set of SPs obtained
      by the flow diagram in Fig. 5 i.e., $\mathcal{S}_{soj} = \left\{(x_j^{soj}, y_j^{soj}) \mid j = 1, 2, ..., \mathcal{N}_{sp}\right\}$
Result: An optimal MS tour sequence
begin
    $t \leftarrow 0$
    $min\_list = [\,]$ ; $min\_len = [\,]$
    for $it = 1$ to $iteration\_threshold$ do
        for each edge $(i, j) \in E$ do
            $\tau_{ij}(t) \leftarrow c$ where $c > 0$ ; $\Delta\tau_{ij} \leftarrow 0$
        end
        Place $l$ ants on $\mathcal{N}_{sp}$ sojourn points
        for $k = 1$ to $l$ do
            $sp\_list^k = [\,]$                    // for each ant $k$ initialize $sp\_list^k$
            $sp\_list^k.append(start^k)$      // insert the starting SP of ant $k$ to the
              first index
        end
        for $k = 1$ to $l$ do
            for $sp = 1$ to $(\mathcal{N}_{sp} - 1)$ do      // number of unvisited sojourn points
                Move ant $k$ to SP $j$ based on $P_{ij}^k(t)$ using Eq. 11
                $sp\_list^k.append(j)$       // insert the visited SP to the next index
            end
            Calculate $L_k$
            Find $min_k L_k$ ,   $k = 1, 2, ...l$
            $min\_len.append(L_k)$ ; $min\_list.append(sp\_list^k)$
        end
        for each edge $(i, j) \in E$ do
            Update $\Delta\tau_{ij}$ using Eq. 13
            Compute $\tau_{ij}(t + \mathcal{N}_{sp})$ using Eq. ??
        end
        if $min\_len[it] == min\_len[it - 1]$ then
            break
        end
    end
    Print $min\_list[it]$
end
```

4.3 Phase 3: steady phase

At the beginning of the data transmission, each non-CH node forwards its sensed data to the respective CH obeying the TDMA schedule. The data aggregation is performed afterwards at the CH level to mitigate the redundant data overhead. In order to avoid collision with other clusters' data, each CH employs a CDMA strategy to forward the aggregated data to the MS. On the other hand, following the aforementioned constructed path (in Algorithm 1) the MS moves to the j^{th} Sojourn Point (SP) for data collection and broadcasts a WAKE_UP MSG (round No, pause time) to wake up all the surrounding CHs which are at a one-hop distance. Consequently, a CH may receive duplicate WAKE_UP MSG if it is located at a one-hop distance from more than one SPs. This issue can be resolved by the field "round number" in the WAKE_UP MSG as the duplicate messages having the same round number are thereby dropped by such CHs. Furthermore, a sojourn time is given for the MS in each SP to gather the CH data, and as soon as the sojourn time expires, the MS moves to the next SP with a certain speed. This process continues until all the SPs are visited by the MS, which eventually transfers the gathered data to the base station for further processing.

5 Analysis of the proposed protocol

Definition 1 SET-COVER. *Given a set system* $\sum = (\mathcal{U}, \mathcal{F})$, *where* $\mathcal{U} = \left\{u_1, \ldots, u_n\right\}$ *is a n element set aka universe and* $\mathcal{F} = \left\{f_1, \ldots, f_m\right\}$ is a family of m subsets such that $\bigcup_i f_i = \mathcal{U}$. The goal is to find the smallest number of these subsets whose union covers all the elements of $\mathcal{U}$.

However, SET-COVER problem is an NP-Hard problem that fails to produce an optimal solution in polynomial time. Hence, the greedy set cover heuristic is heavily exploited for generating a poynomial time solution with an approximation factor of $\ln n$. The proposed MS sojourn point nomination algorithm is a SET-COVER problem where $\mathcal{U} = \mathcal{S}_{CH}$ and our aim is to obtain a minimal cardinality set $\mathcal{S}_{soj}$ of sojourn points that satisfies the constraint 9b. In this context, each nominated SP $i \in \mathcal{S}_{soj}$ results in a subset of cluster heads $\mathcal{S}_i^{sub}$

such that $\bigcup_i \mathcal{S}_i^{sub} = \mathcal{S}_{CH}$ i.e. $\mathcal{F} = \left\{\mathcal{S}_1^{sub}, \dots, \mathcal{S}_i^{sub}, \dots, \mathcal{S}_{\mathcal{N}_{sp}}^{sub}\right\}$ where $\left|\mathcal{S}_{soj}\right| = \mathcal{N}_{sp}$.

Theorem 1 *The proposed SP nomination algorithm takes at most* $(\mathcal{N}'_{sp} \ln \mathcal{N}_{CH} + 1)$ *number of iterations to cover* $\mathcal{S}_{CH}$ *where* $\mathcal{N}'_{sp}$ *is the minimal number of SPs used by the optimal solution.*

Proof Let us assume the following notations to analyze the performance of the proposed greedy heuristic:

- $\mathcal{N}'_{sp}$ denotes the number of SPs used by the optimal solution to cover $\mathcal{S}_{CH}$.
- $\mathcal{N}_{sp}$ denotes the number of SPs used by the greedy heuristic to cover $\mathcal{S}_{CH}$.
- a_i denotes the number of new CHs covered at iteration i.
- b_i denotes the number of CHs remain uncovered at the end of iteration i.

We will begin with $b_0 = n$ and at the 1^{st} iteration, we cover a_1 number of new CHs that results in $b_1 = \mathcal{N}_{CH} - a_1$, the number of uncovered CHs. In the 2^{nd} iteration, we will cover a_2 number of another CHs that results in $b_2 = b_1 - a_1$, and so on. Now, if the optimal solution is covering all the CHs with $\mathcal{N}'_{sp}$ number of SPs, then there must a subset $\mathcal{S}_i^{sub} \in \mathcal{F}$ covering at least $\frac{\mathcal{N}_{CH}}{\mathcal{N}'_{sp}}$ number of CHs. In addition, the proposed greedy heuristic picks the biggest set of CHs in each iteration and hence, covers at least that many CHs. At any iteration i, the amount of CHs covered by the greedy heuristic is at least as good as the remaining CHs divided by the $\mathcal{N}'_{sp}$ i.e., $a_i \geq \frac{b_{i-1}}{\mathcal{N}'_{sp}}$. Hence, the expression for the number of CHs uncovered at i_{th} iteration can be expressed as:

$$\begin{aligned} b_i &= b_{i-1} - a_i \\ &\leq b_{i-1} - \frac{b_{i-1}}{\mathcal{N}'_{sp}} = \left(1 - \frac{1}{\mathcal{N}'_{sp}}\right) b_{i-1} \end{aligned} \tag{14}$$

According to the mathematical induction,

$$b_i \leq b_0 \left(1 - \frac{1}{\mathcal{N}'_{sp}}\right)^i = \mathcal{N}_{CH}\left(1 - \frac{1}{\mathcal{N}'_{sp}}\right)^i \tag{15}$$

After $\mathcal{N}'_{sp}$ iterations nominating $\mathcal{N}'_{sp}$ number of SPs, the number of leftover CHs are:

$$b_{\mathcal{N}'_{sp}} = \mathcal{N}_{CH}\left(1 - \frac{1}{\mathcal{N}'_{sp}}\right)^{\mathcal{N}'_{sp}} \leq \mathcal{N}_{CH}\left(\frac{1}{e}\right) \tag{16}$$

[by Taylor's expansion]

Hence, the proposed greedy heuristic will have no more than $\mathcal{N}_{CH} \times \left(\frac{1}{e}\right)$ number of CHs left at $\mathcal{N}'_{sp}{}^{th}$ iteration. If we put $\mathcal{N}_{sp} = \mathcal{N}'_{sp} \times ln\mathcal{N}_{CH}$, then

$$b_{\mathcal{N}_{sp}} = \mathcal{N}_{CH}\left(1 - \frac{1}{\mathcal{N}'_{sp}}\right)^{\mathcal{N}'_{sp} \times \ln \mathcal{N}_{CH}} \leq \mathcal{N}_{CH} \times e^{-\ln \mathcal{N}_{CH}} = 1 \tag{17}$$

This states after $\mathcal{N}_{sp} = \mathcal{N}'_{sp} \times ln\mathcal{N}_{CH}$ iterations, the proposed heuristic will have atmost one CH left. In other words, the heuristic will terminate in no more than $(\mathcal{N}'_{sp} \ln \mathcal{N}_{CH} + 1)$ iterations which proves the Theorem 1. □

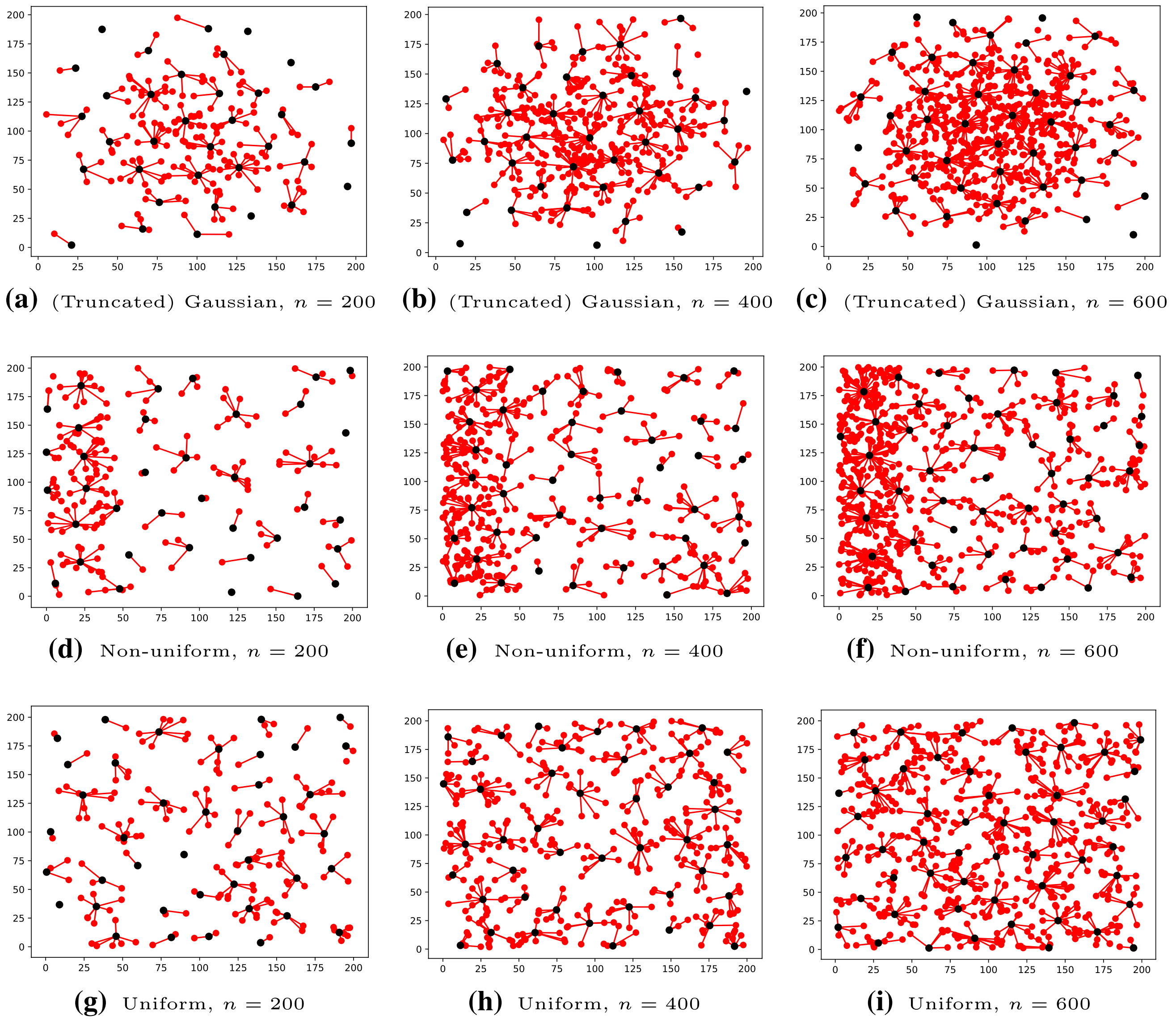
Theorem 2 *The proposed MS traversal path construction algorithm guarantees a solution of minimal length tour converging in polynomial time.*

Proof The proposed MS traversal path construction algorithm is similar to $\mathcal{N}_{sp}$ city TSP where the objective is minimize the tour distance (Eq. 10a). Such a hard combinatorial optimization problem involves $\frac{(\mathcal{N}_{sp}-1)!}{2}$ number of MS tour possibilites where $\mathcal{S}_{can} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_{\frac{(\mathcal{N}_{sp}-1)!}{2}}\}$ is the set of all possible candidate solutions. Every candidate solution $\mathcal{C}_i \in \mathcal{S}_{can}$ is one sequence of sojourn points traversed by the MS in a given order. Among them, the optimal MS traversal path $\mathcal{C}_{OPT} \in \mathcal{S}_{can}$ ought to be discovered in polynomial time.

Herein, we have applied ACO algorithm including a fully connected graph $G = (\mathcal{S}_{soj}, \mathcal{E}_{soj})$ and m ants where after $\left(\mathcal{N}_{sp} \times m\right)$ number of iterations the trail intensity of every edge $(i,j) \in \mathcal{E}_{soj}$ is updated. The pheromone updation (Eq. 12) of ACO is formulated in such a way that the shorter edges will have higher pheromone value (the number of ants visited the respective edge). Hence, after $\left(\mathcal{NC}_{max} \times \mathcal{N}_{sp} \times m\right)$ number of iterations the path possessing highest number of pheromone trail intensity is nominated as the optimal MS path (minimal distance path) where $\mathcal{NC}_{max}$ is the maximum value of the cycle counter. In other words, ACO will generate an minimal length MS tour and stops iterating when $\mathcal{NC}_i = \mathcal{NC}_{max}$ or the convergence criteria is met which proves the Theorem 2. □

6 Performance evaluation

In practice, the implementation of WSN routing protocols is quite expensive and arduous for large-scale applications. This necessitates the use of simulation tools to analyze and evaluate the performance of the WSN protocols. In this section, a thorough simulation analysis is carried out to manifest the improvements of the proposed protocol over the



(a) (Truncated) Gaussian, $n = 200$ **(b)** (Truncated) Gaussian, $n = 400$ **(c)** (Truncated) Gaussian, $n = 600$

(d) Non-uniform, $n = 200$ **(e)** Non-uniform, $n = 400$ **(f)** Non-uniform, $n = 600$

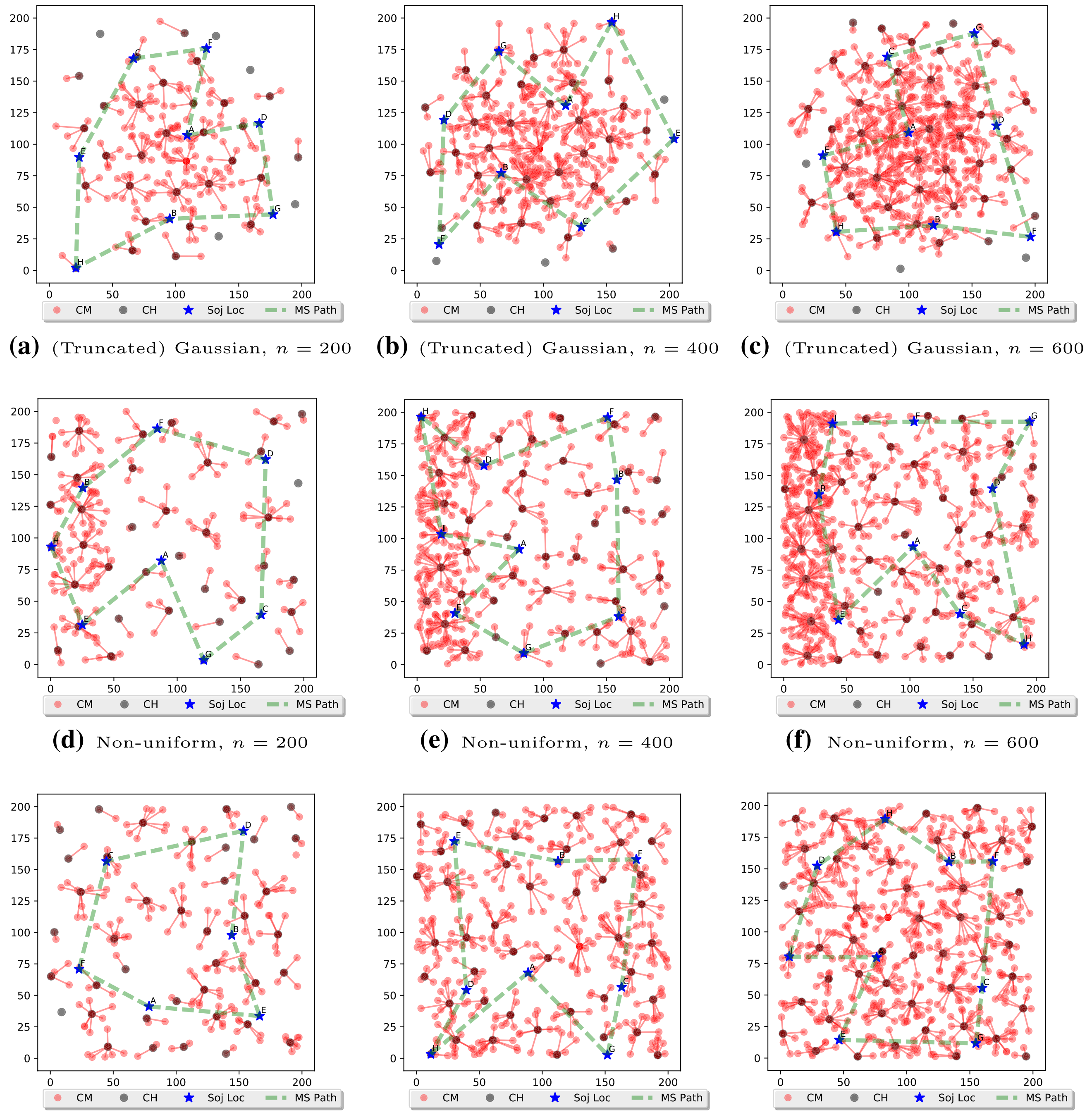
(g) Uniform, $n = 200$ **(h)** Uniform, $n = 400$ **(i)** Uniform, $n = 600$

Fig. 13 Clustered network for different WSN deployments with varying node densities

related existing protocols like EAPC (Wen et al. 2017b), Dynamic (Krishnan et al. 2018a), and Distributed (Alsaafin et al. 2018). The whole set of simulations is conducted using Python programming language (NetworkX Python library) with the Spyder 3.1.2 development environment running in an octa-core Intel(R) processor, Xeon(R) server on the Windows Server 2012R2 operating system.

6.1 Simulation environment

The simulation environment assumes a number of sensor nodes having initial energy $E_{init} = 0.5$ Jule are deployed over a target area of size $200 \times 200\, m^2$. As mentioned in Sect. 3.1, this article considers three distinct WSN deployments following (truncated) Gaussian, non-uniform, and uniform distributions that demonstrates the flexibility of our protocol in different WSN applications. Moreover, it has been noticed from empirical studies that network performance is substantially influenced by varying the denity of the sensor nodes. Therefore, this article accomplishes the simulation analysis of the studied protocols in all considered WSN deployments with different number of nodes (n). Figure 1 shows several initial deployment scenarios where Fig. 1a–c exhibit the (truncated) Gaussian-distributed WSNs for 200, 400, and 600 number of nodes and Fig. 1d–i exhibit the same for non-uniform and uniformly distributed WSNs, respectively. The corresponding clustered networks are shown in Fig. 13a–i while Fig. 14a–i exhibit the shortest possible MS trajectory for collecting CH data from optimal set of sojourn points.



(a) (Truncated) Gaussian, $n = 200$ **(b)** (Truncated) Gaussian, $n = 400$ **(c)** (Truncated) Gaussian, $n = 600$

(d) Non-uniform, $n = 200$ **(e)** Non-uniform, $n = 400$ **(f)** Non-uniform, $n = 600$

(g) Uniform, $n = 200$ **(h)** Uniform, $n = 400$ **(i)** Uniform, $n = 600$

Fig. 14 Optimal MS trajectory for different WSN deployments with varying node densities

6.2 Energy model

In the proposed architecture, we use the radio energy dissipation model (Heinzelman et al. 2000) to estimate the energy consumption of the sensor nodes during data transmission and aggregation. According to the first order radio model, for transmitting b bits of data over a distance of δ, the radio expends:

$$E_T(b, \delta) = b \times \varepsilon_{elec} + b \times \varepsilon_{amp} \times \delta^e \tag{18}$$

where ε_{elec} and ε_{amp} are the energy dissipation per bit in the electronic circuit and the amplifier, respectively. The path loss exponent is denoted by e that relies on the transmission distance δ. When δ is less than the threshold distance δ_{th}, the free space (*fs*) model is applied to set the *e* value to 2 and $\varepsilon_{amp} = \varepsilon_{fs}$ otherwise, the multipath (*mp*) model is applied to set *e* to 4 and $\varepsilon_{amp} = \varepsilon_{mp}$. The threshold distance is derived as $\delta_{th} = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}}$. Hence, the transmission energy Eq. 18 can be rewritten as

$$E_T(b,\delta) = \begin{cases} b \times \varepsilon_{elec} + b \times \varepsilon_{fs} \times \delta^2 & \text{if } \delta < \delta_{th} \\ b \times \varepsilon_{elec} + b \times \varepsilon_{mp} \times \delta^4 & \text{if } \delta \geq \delta_{th} \end{cases} \tag{19}$$

Similarly, to receive and aggregate a data packet of b bits radio expends respectively $E_R(b) = b \times \varepsilon_{elec}$ and $E_{agg} = b \times \varepsilon_{da}$ where ε_{da} is the dissipated energy for one bit data aggregation. For the experimental purpose, the radio model parameter values are taken as $\varepsilon_{elec} = 50\, nJ$, $\varepsilon_{fs} = 10\, pJ/bit/m^2$, $\varepsilon_{mp} = 0.0013\, pJ/bit/m^4$, and $\varepsilon_{da} = 5\, nJ/bit/signal$. In addition, the transmitted data and control packets are considered of size 4000 bits and 200 bits respectively.

6.3 Performance metrics:

The efficacy of the proposed protocol is measured with respect to the following performance metrics:

- Energy consumption (in Joule): In the proposed model, energy consumption of the sensor nodes is caused by the data and control packet exchanges during set-up, routing, and steady phases. The total energy consumed by the network in the first round can be calculated as:

$$E^1_{con} = \sum_{i=1}^{n} e^i_1$$

where e^i_1 is the energy consumed by the i^{th} sensor node in first round and n is the total number of sensor nodes. The following rounds evaluate the network energy consumption as a cumulative sum of the current and earlier rounds' energy consumption. For example, total energy consumption in $\mathcal{R}^{th}$ round can be expressed as (Roy et al. 2020):

$$E^r_{con} = \sum_{i=1}^{n} e^i_1 + \sum_{i=1}^{n} e^i_2 + \cdots \ \mathcal{R}\ times \tag{20}$$

- Number of alive nodes: It indicates the total number of sensor nodes possessing higher residual energy than the energy threshold $\mathcal{E}$ (minimum energy needed by any sensor node to accomplish the network operation) (Roy et al. 2020). Therefore, in a round $\mathcal{R}$, a sensor node i is claimed to be alive if

$$\left(E_{init} - \sum_{r=1}^{\mathcal{R}-1} e^i_r\right) > \mathcal{E} \quad i.e., \quad E^i_{res} > \mathcal{E} \tag{21}$$

where E_{init} is the initial energy of every sensor node and E^i_{res} is the residual energy of i^{th} sensor node.

- First node die (FND): It denotes the number of rounds after which first alive node dies (Roy et al. 2020) i.e. its residual energy falls below the energy threshold $\mathcal{E}$. The mathematical expression of FND is as follows:

$$FND = min\left\{\mathcal{R}^i \mid i = 1, 2, \ldots, n\right\}$$
$$\text{subject to} \quad \left(E_{init} - \sum_{r=1}^{\mathcal{R}^i} e^i_r\right) \leq \epsilon \tag{22}$$

where $\mathcal{R}^i$ is the lifetime of sensor node i.

- Packet delivery ratio (PDR): It is the proportion of the total number of data packets reached to MS ($\mathcal{P}^i_{ms}$) to the total number of packets transmitted i.e.,

$$PDR = \frac{\mathcal{P}^i_{MS}}{\sum_{i=1}^{n} \mathcal{P}^i_{gen}} \tag{23}$$

where $\mathcal{P}^i_{gen}$ is the is the number of packets generated at i^{th} senseor node. In real life applications, packets received at the MS are always lesser in number than packets transmitted due to error-prone medium, heavy congestion, hostile environment etc. According to the random uniformed model (Abo-Zahhad et al. 2015) packet loss probability between a sender p and receiver q is defined by

$$P_{loss}(p,q) = \begin{cases} 0 & \delta(p,q) \in [0, 50) \\ 0.01 * (\delta(p,q) - 50) & \delta(p,q) \in [50, 100] \\ 1 & \delta(p,q) \in (100, \infty) \end{cases} \tag{24}$$

This implies that the packet is assumed to be successfully delivered when the link probability between the sender and receiver is greater than $P_{loss}(p,q)$ otherwise, it is dropped. In other words, packet loss probability increases dynamically with the increase in the transmission distance.

- End-to-end delay (E2Ed): This metric is referred to the time (in ms) required for a packet generated by some sensor node to be successfully delivered to the sink (Roy et al. 2020). It involves transmision, propagation and queuing delays. The average E2Ed of the network is estimated as:

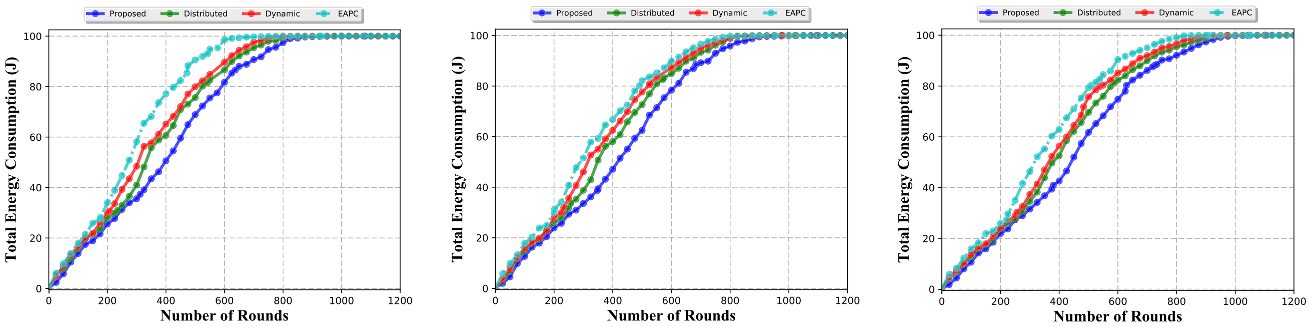
$$E2Ed_{avg} = \frac{\sum_{i=1}^{n} \mathcal{P}^i_{gen} \times (t_a - t_g)^i}{\sum_{i=1}^{n} \mathcal{P}^i_{gen}} \tag{25}$$

where t_g is single packet generation time at i^{th} sensor node, t_a is packet arrival time at the MS, $\mathcal{P}^i_{gen}$ is the number of packets generated at i^{th} senseor node and sucessfully delivered to the sink.

- Average hop count (AHC): Hop count of a source node indicates the number of intermediate nodes through which its data will be forwarded to the destination node. AHC measures the average of all CHs' hop count in the network to relay their cluster data to the MS.

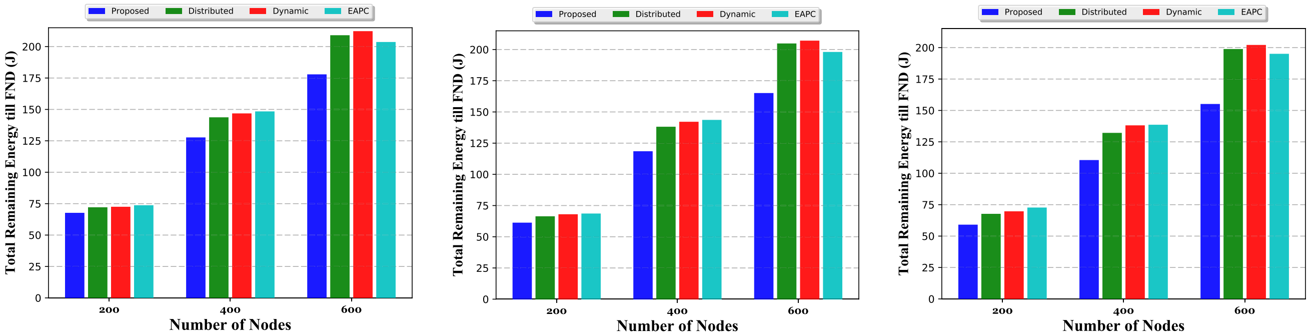
6.3.1 Energy efficiency

Conserving sensor energy is a paramount concern for any energy-constrained WSN application as it refers to the longevity of the network. Hence, a comparison graph



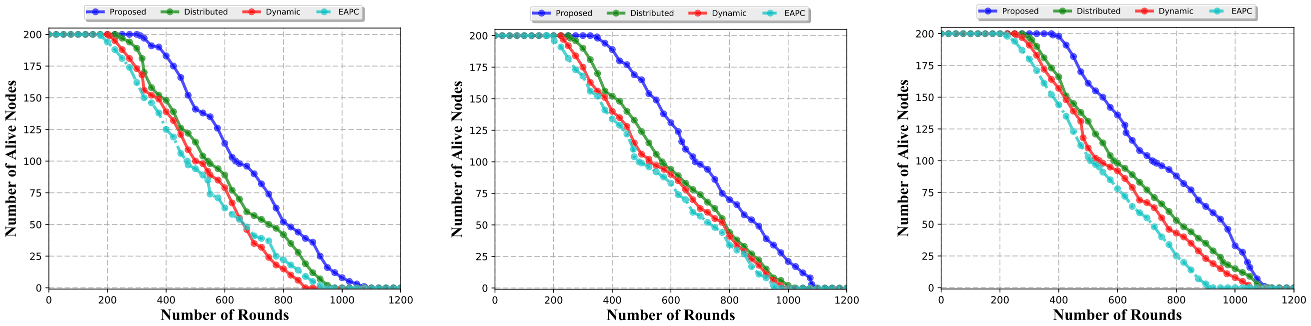
(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 15 Total energy consumption for different WSN deployments where $n = 200$



(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 16 Total remaining energy till FND for different WSN deployments with varying node densities (n = 200, 400, and 600)



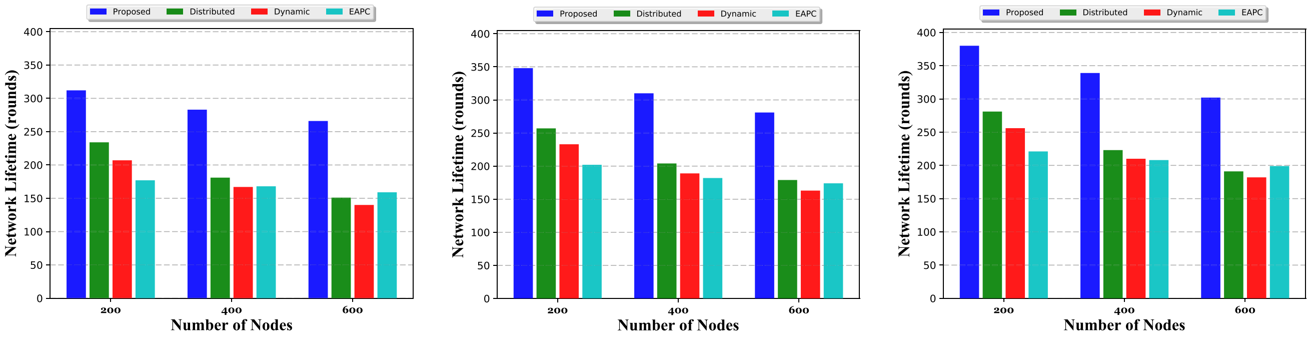
(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 17 Number of alive nodes for different WSN deployments where $n = 200$

is presented in Fig. 15 among the proposed protocol and existing benchmarks in terms of total energy consumption (as defined in Eq. 20) for 200 number of deployed sensors. This graph exhibits the dominance of our protocol over the others as it achieves minimal energy consumption in each round by ensuring one-hop data delivery to the sink along with a limited number of Sojourn points (SP). On the other hand, EAPC (Wen et al. 2017b) encounters multi-hop data forwarding strategy which leads to higher energy consumption during data communication whereas dynamic (Krishnan et al. 2018a) and distributed (Alsaafin et al. 2018) suffer from higher transmission energy expenditure despite assuring one-hop data delivery. This is because of their large number of data collection points which increases the MS traversal latency and eventually may lead to buffer overflow of the CHs. Subsequently, re-transmission of the dropped packets will further expend the energy of the network. However, it is of great importance to manifest the efficacy of the proposed protocol under the impact of varying node densities in order to guarantee its scalability. Hence, Fig. 16 depicts bar representations of the four studied protocols in respect of total remaining energy till FND for different node densities (200–600). It is obvious that the total remaining energy till FND for the proposed one is much lower than the existing ones, which approves a more balanced energy consumption among the sensors.

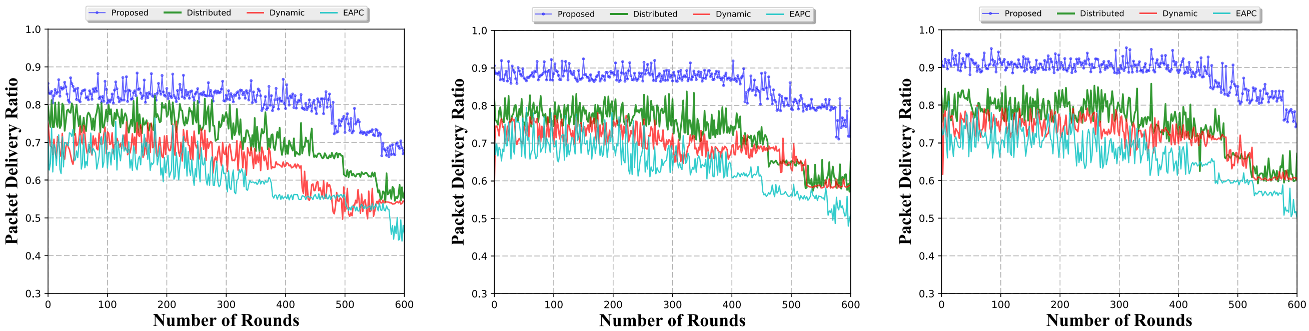
6.3.2 Network lifetime

An extended network lifetime is always desirable in energy-limited WSN applications. In this regard, Fig. 17 exhibits a comparative analysis among the foregoing protocols with respect to the number of alive nodes (as defined in Eq. 21). The comparison is made for 200 number of deployed nodes under (truncated) Gaussian, non-uniform, and uniformly distributed scenarios. It is clear that the proposed protocol achieves substantial enhancement in prolonging the network lifetime over the related protocols. Enhanced network lifetime is achieved due to the minimal energy consumption in each network round caused by the one-hop data transmission to the MS sojourn points. Herein, the terminology for network lifetime is set as FND (Eq. 22). However, evaluating the network performance by means of FND may not be an ideal measure as most of the WSN protocols function satisfactorily even after a certain number of sensor nodes die. Hence, the line graphs (Fig. 17) regarding the number



(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 18 Network lifetime for different WSN deployments with varying node densities (n = 200, 400, and 600)



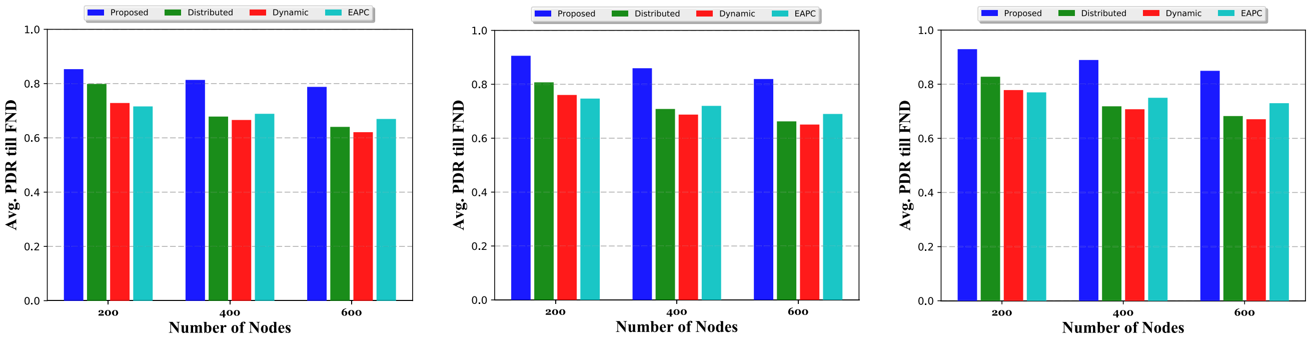
(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 19 Packet delivery ratio for different WSN deployments where n = 200

of alive nodes consider the comparative analysis among the proposed and related existing ones until the last node dies. From Fig. 18a, we can observe that the proposed protocol improves the network lifetime in case of (truncated) Gaussian deployment by 33–76% × than Distributed (Alsaafin et al. 2018), 51–90% × than dynamic (Krishnan et al. 2018a), and 67–76% × than EAPC (Wen et al. 2017b). From Fig. 18b we can further observe the improvement of network lifetime in case of non-uniform deployment by 35–57% × than distributed, 49–72% × than dynamic, and 61–72% × than EAPC protocol while Fig. 18c exhibits the same for uniform deployment by 35–58% × than distributed, 48–66% × than dynamic, and 52–72% × than EAPC protocol. The consistent performance gain of the proposed protocol in all considered scenarios over the compared protocols validates its better scalability.

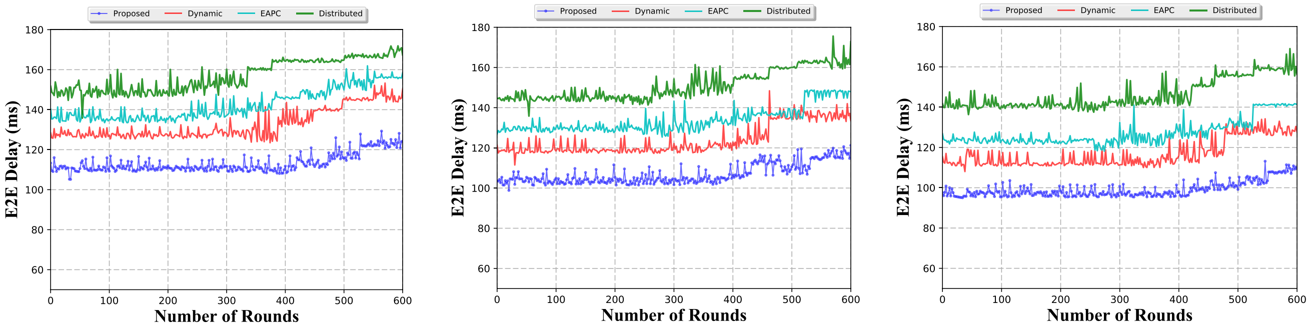
6.3.3 Packet delivery

In addition to the network lifetime, network QoS metrics like packet delivery ratio and end-to-end delay are also essential to assess the effectiveness of the WSN routing protocols. As defined in Eq. 23, PDR is the rate of received packets where higher PDR value refers to the better end-to-end reliability of the network. However, in practice, sharing an error-prone medium, hazardous environment, and heavy network traffic may lead to packet loss during data transmission; accordingly, reduces the network end-to-end reliability. Eqution 24 states that packet loss probability is directly proportional to the propagation distance between sender and receiver. Figure 19 reveals that the proposed protocol outperforms the related existing studies with respect to PDR for $n = 200$. EAPC (Wen et al. 2017b) suffers from severe packet drop after HND due to the long haul multi-hop communication in the unbalanced scenario. On the other side, dynamic (Krishnan et al. 2018a), distributed (Alsaafin et al. 2018), and the proposed one suggest a one-hop data delivery to the MS to ensure a higher PDR. However, unlike the proposed protocol, distributed and dynamic both recommend a substantially large number of data collection points which results in buffer overflow at the CHs, and consequently degrades their performance. Moreover, the bar graphs in Fig. 20 validate the competence and scalability of the proposed protocol for all considered scenarios by presenting the average PDR value till FND for variable node densities (200–600). From 20a, it is noticed that the proposed



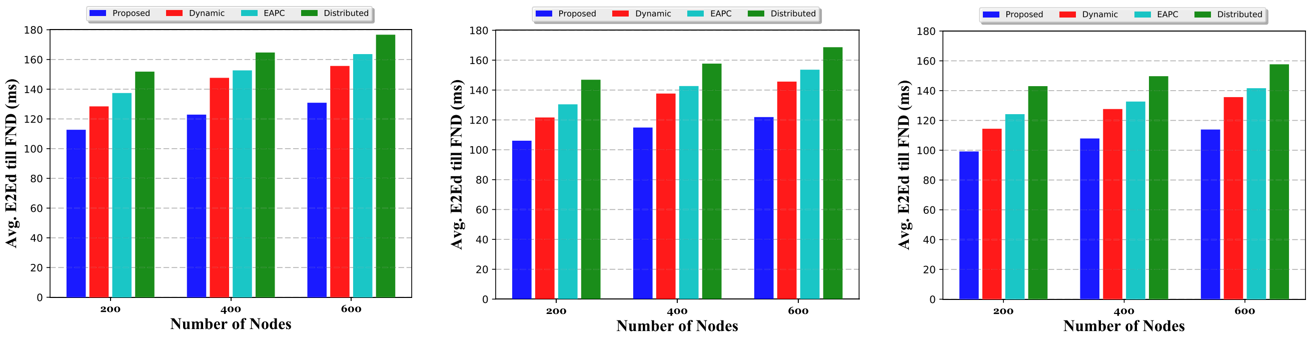
(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 20 Average Packet Delivery Ratio (PDR) till FND for different WSN deployments with varying node densities ($n = 200$, 400, and 600)



(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 21 End-to-end delay (E2Ed) for different WSN deployments where $n = 200$



(a) (Truncated) Gaussian **(b)** Non-uniform **(c)** Uniform

Fig. 22 Average E2Ed till FND for different WSN deployments with varying node densities (n = 200, 400, and 600)

protocol increases the average PDR till FND by 7–23% × than distributed, 17–27% × than dynamic, and 17–19% × than EAPC. From Fig. 20b, we can observe that the proposed protocol raises the average PDR till FND by 12–24% × than distributed, 19–26% × than dynamic, and 18–21% × than EAPC. We can further notice from Fig. 20c, that the proposed one improves the average PDR till FND by 12–24% × than distributed, 19–27% × than dynamic, and 16–21% × than EAPC.

6.3.4 End-to-end delay

Another crucial QoS routing metric, end-to-end delay (defined in Eq. 25) of the transmitted data packets is also taken into account herein to evaluate the studied protocols. A lower value of E2Ed is always desirable in delay-sensitive WSN applications as it refers to the data delivery timeliness. A comparison graph for $n = 200$ is depicted in Fig. 21 that demonstrates the superiority of our protocol over the compared ones with respect to the average end-to-end delay of the network. Both EAPC (Wen et al. 2017b) and dynamic (Krishnan et al. 2018a) experience higher E2Ed because of the presence of long-haul multi-hop routing. On the other hand, distributed (Alsaafin et al. 2018) despite offering a one-hop data gathering scheme, suffers from high E2Ed as it lengthens the MS traversal path comprising a larger number of sojourn points. The proposed protocol suggests a one-hop data gathering with the minimal possible number of sojourn points and attains a considerably lower E2Ed. In addition, Fig. 22 depicts the bar representations of the foregoing protocols with respect to average E2Ed till FND for $n = 200$, 400, and 600, respectively. It is noticed from Fig. 22a that the proposed protocol mitigates the average E2Ed value till FND by 14–20% × than dynamic, 22–25% × than EAPC, and 34–35% × than Distributed. Likewise, Fig. 22b shows that the average delay of the proposed protocol till FND is improved by 15–20% × than dynamic, 22–26% × than EAPC, and 37–39% × than distributed. It is further observed from Fig. 22c that the average delay of the proposed protocol till FND is reduced by 15–19% × than dynamic, 23–25% × than EAPC, and 38–44% × than Distributed. It is noteworthy that an increase in node density increases the congestion in the network, which inflicts higher end-to-end delay.

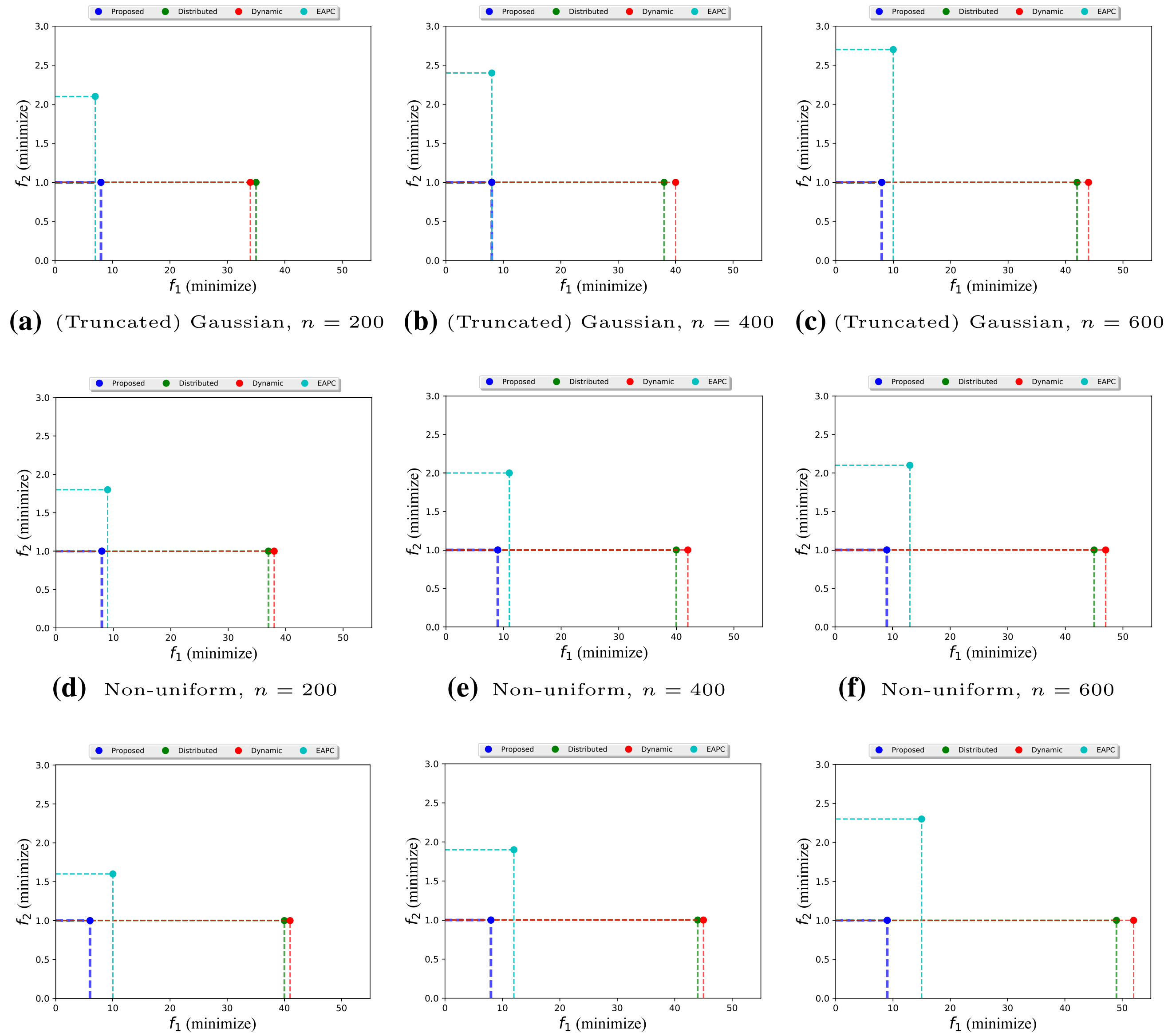
6.3.5 Hop count and sojourn points

It has been noticed that higher average hop count (as defined in Sect. 6.3) value of a WSN increases the likelihood of the hotspot problem. This enforces the proposed protocol to focus on minimizing the AHC value of the network. In addition to the proposed one, dynamic (Krishnan et al. 2018a) and distributed (Alsaafin et al. 2018) both obtain a one-hop data gathering which produces the least AHC value. On the contrary, EAPC (Wen et al. 2017b) encounters the multi-hop forwarding and suffers from hotspot issue. However, minimizing the hop count may increase the number of MS sojourn points ($\mathcal{N}_{sp}$) which may not secure the delay-sensitive applications of WSN. This introduces nothing but a trade-off issue between AHC and $\mathcal{N}_{sp}$ that can be formulated as:

$$\left.\begin{array}{ll} \text{minimize} & f_1(\mathcal{S}_{CH}, \mathcal{S}_{CH}^{loc}) = AHC \\ \text{minimize} & f_2(\mathcal{S}_{CH}, \mathcal{S}_{CH}^{loc}) = \mathcal{N}_{sp} \\ \text{subject to} & \left|\mathcal{S}_{CH}\right| \qquad\qquad < < |\mathcal{S}| \\ \multicolumn{2}{l}{0 \le x_i^{ch} \in \mathcal{S}_{CH}^{loc} \le 200 \;\; ; \;\; 0; \le y_i^{ch} \in \mathcal{S}_{CH}^{loc} \le 200} \end{array}\right\} \tag{26}$$

where f_1 and f_2 are two conflicting objective functions, $\mathcal{S}_{CH}$ is the set of all CHs, and $\mathcal{S}_{CH}^{loc}$ is the set of all CH coordinates i.e. $\mathcal{S}_{CH}^{loc} = \left\{(x_i^{ch}, y_i^{ch}) \mid i = 1, 2, \dots, \mathcal{N}_{CH}\right\}$. Figure 23a–i exhibit four solutions of the minimization problem 26 in different deployment scenarios where each solution is obtained by the proposed and compared existing protocols at any randomly selected round.

However, the proposed solution achieves a great balance between both the objectives (f1 and f2), which clearly



(a) (Truncated) Gaussian, $n = 200$ **(b)** (Truncated) Gaussian, $n = 400$ **(c)** (Truncated) Gaussian, $n = 600$

(d) Non-uniform, $n = 200$ **(e)** Non-uniform, $n = 400$ **(f)** Non-uniform, $n = 600$

(g) Uniform, $n = 200$ **(h)** Uniform, $n = 400$ **(i)** Uniform, $n = 600$

Fig. 23 A population of four solutions for different WSN deployments where $n = 200$

manifests its efficacy in both energy-aware and delay-sensitive applications.

7 Conclusion

In this paper, a novel mobile sink based data gathering protocol has been proposed for a clustered wireless sensor network to mitigate the energy hole problem. The proposed protocol aims to nominate an optimal set of SPs such that each SP is surrounded by the maximum possible number of cluster heads in one-hop proximity. Afterwards, the shortest possible MS traversal route is established in polynomial time to visit all the nominated SPs by means of the ACO algorithm. Thus, the proposed protocol attains an energy efficiency by ensuring one-hop communication between the CHs and MS, while the data collection latency is minimized by designing a minimal length path to cover the SPs. More concretely, this article provides an effective balance between energy consumption and data collection latency in the network. The performance evaluation exhibits that the proposed protocol consistently outperforms the related existing literature in (truncated) Gaussian, non-uniform, and uniformly distributed deployments with various node densities. This manifests the viability of the proposed protocol in different scale WSNs. In the future, this protocol can be further extended to multiple MS-based data gathering problems under storage as well as time-constrained WSN applications.

References

Abo-Zahhad M, Ahmed SM, Sabor N, Sasaki S (2015) Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks. IEEE Sens J 15(8):4576–4586

Akkari W, Bouhdid B, Belghith A (2015) Leatch: low energy adaptive tier clustering hierarchy. Proc Comput Sci 52:365–372

Alaei M, Yazdanpanah F (2019) Eelcm: an energy efficient load-based clustering method for wireless mobile sensor networks. Mob Netw Appl 24(5):1486–1498

Alsaafin A, Khedr AM, Al Aghbari Z (2018) Distributed trajectory design for data gathering using mobile sink in wireless sensor networks. AEU Int J Electron Commun 96:1–12

Benzaïd C, Bagaa M, Younis M (2017) Efficient clock synchronization for clustered wireless sensor networks. Ad Hoc Netw 56:13–27

Chen DR, Chen LC, Chen MY, Hsu MY (2019) A coverage-aware and energy-efficient protocol for the distributed wireless sensor networks. Comput Commun 137:15–31

Dorigo M, Stützle T (2019) Ant colony optimization: overview and recent advances. In: Handbook of metaheuristics. Springer, pp 311–351

Gajjar S, Sarkar M, Dasgupta K (2016) Famacrow: fuzzy and ant colony optimization based combined mac, routing, and unequal clustering cross-layer protocol for wireless sensor networks. Appl Soft Comput 43:235–247

Ghafoor S, Rehmani MH, Cho S, Park SH (2014) An efficient trajectory design for mobile sink in a wireless sensor network. Comput Electr Eng 40(7):2089–2100

Gherbi C, Aliouat Z, Benmohammed M (2016) An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks. Energy 114:647–662

Hamidouche R, Aliouat Z, Gueroui AM (2018) Genetic algorithm for improving the lifetime and qos of wireless sensor networks. Wirel Pers Commun 101(4):2313–2348

Harizan S, Kuila P (2020) A novel nsga-ii for coverage and connectivity aware sensor node scheduling in industrial wireless sensor networks. Dig Signal Process 102753

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

Heinzelman WR, Chandrakasan A, Balakrishnan H (2000) Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd annual Hawaii international conference on system sciences. IEEE, p 10

Huang H, Savkin AV (2017) An energy efficient approach for data collection in wireless sensor networks using public transportation vehicles. AEU Int J Electron Commun 75:108–118

Huang H, Savkin AV, Huang C (2017) I-umdpc: the improved-unusual message delivery path construction for wireless sensor networks with mobile sinks. IEEE Internet Things J 4(5):1528–1536

Huang H, Savkin AV, Ding M, Huang C (2019) Mobile robots in wireless sensor networks: a survey on tasks. Comput Netw 148:1–19

Kaur T, Kumar D (2018) Particle swarm optimization-based unequal and fault tolerant clustering protocol for wireless sensor networks. IEEE Sens J 18(11):4614–4622

Khedr AM, Osamy W, Salim A (2018) Distributed coverage hole detection and recovery scheme for heterogeneous wireless sensor networks. Comput Commun 124:61–75

Krishnan M, Yun S, Jung YM (2018a) Dynamic clustering approach with aco-based mobile sink for data collection in wsns. Wirel Netw 1–13

Krishnan M, Yun S, Jung YM (2018b) Improved clustering with firefly-optimization-based mobile data collector for wireless sensor networks. AEU Int J Electron Commun 97:242–251

Krishnan M, Yun S, Jung YM (2019) Enhanced clustering and aco-based multiple mobile sinks for efficiency improvement of wireless sensor networks. Comput Netw 160:33–40

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

Lai WK, Fan CS, Lin LY (2012) Arranging cluster sizes and transmission ranges for wireless sensor networks. Inf Sci 183(1):117–131

Logambigai R, Ganapathy S, Kannan A (2018) Energy-efficient grid-based routing algorithm using intelligent fuzzy rules for wireless sensor networks. Comput Electr Eng 68:62–75

Mazumdar N, Om H (2018) Distributed fuzzy approach to unequal clustering and routing algorithm for wireless sensor networks. Int J Commun Syst 31(12):e3709

Mehto A, Tapaswi S, Pattanaik K (2020) Pso-based rendezvous point selection for delay efficient trajectory formation for mobile sink in wireless sensor networks. In: 2020 International conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, pp 252–258

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

Qin Z, Wu D, Xiao Z, Fu B, Qin Z (2018) Modeling and analysis of data aggregation from convergecast in mobile sensor networks for industrial iot. IEEE Trans Ind Inform 14(10):4457–4467

Ramos HS, Boukerche A, Oliveira AL, Frery AC, Oliveira EM, Loureiro AA (2016) On the deployment of large-scale wireless sensor networks considering the energy hole problem. Comput Netw 110:154–167

Ren J, Zhang Y, Zhang K, Liu A, Chen J, Shen XS (2015) Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks. IEEE Trans Ind Inform 12(2):788–800

Roy S, Mazumdar N, Pamula R (2020) An energy and coverage sensitive approach to hierarchical data collection for mobile sink based wireless sensor networks. J Ambient Intell Humaniz Comput 1–25

Sabet M, Naji HR (2015) A decentralized energy efficient hierarchical cluster-based routing algorithm for wireless sensor networks. AEU Int J Electron Commun 69(5):790–799

Sabor N, Abo-Zahhad M, Sasaki S, Ahmed SM (2016) An unequal multi-hop balanced immune clustering protocol for wireless sensor networks. Appl Soft Comput 43:372–389

Salem AOA, Shudifat N (2019) Enhanced leach protocol for increasing a lifetime of wsns. Pers Ubiquitous Comput 1–7

Shahraki A, Taherkordi A, Haugen Ø, Eliassen F (2020) Clustering objectives in wireless sensor networks: a survey and research direction analysis. Comput Netw 180:107376

Shivalingegowda C, Jayasree P (2020) Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks. Journal of Ambient Intelligence and Humanized Computing pp 1–14

Tang J, Huang H, Guo S, Yang Y (2015) Dellat: delivery latency minimization in wireless sensor networks with mobile sink. J Parallel Distrib Comput 83:133–142

Tao J, He L, Zhuang Y, Pan J, Ahmadi M (2012) Sweeping and active skipping in wireless sensor networks with mobile elements. In: 2012 ieee global communications conference (GLOBECOM). IEEE, pp 106–111

Ukani V, Thakkar P, Parikh V (2019) A range-based adaptive and collaborative localization for wireless sensor networks. In:

Information and communication technology for intelligent systems. Springer, pp 293–302

Wang Y, Fu W, Agrawal DP (2012) Gaussian versus uniform distribution for intrusion detection in wireless sensor networks. IEEE Trans Parallel Distrib Syst 24(2):342–355

Wang Z, Zhang B, Wang X, Jin X, Bai Y (2018) Improvements of multihop localization algorithm for wireless sensor networks. IEEE Syst J 13(1):365–376

Wen W, Dong Z, Chen G, Zhao S, Chang CY (2017a) Energy efficient data collection scheme in mobile wireless sensor networks. In: 2017 31st international conference on advanced information networking and applications workshops (WAINA). IEEE, pp 226–230

Wen W, Zhao S, Shang C, Chang CY (2017b) Eapc: energy-aware path construction for data collection using mobile sink in wireless sensor networks. IEEE Sens J 18(2):890–901

Wen W, Chang CY, Zhao S, Shang C (2018) Cooperative data collection mechanism using multiple mobile sinks in wireless sensor networks. Sensors 18(8):2627

Yu J, Qi Y, Wang G, Gu X (2012) A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution. AEU Int J Electron Commun 66(1):54–61