A robust uncertainty-aware cluster-based deployment approach for WSNs: Coverage, connectivity, and lifespan

Mustapha Reda Senouci, Abdelhamid Mellouk

A robust uncertainty-aware cluster-based deployment approach for WSNs: coverage, connectivity, and lifespan

Mustapha Reda Senouci, Abdelhamid Mellouk, *Senior Member, IEEE*

Abstract

Predicting the expected performance of Wireless Sensor Networks (WSNs) is the key to their successful deployment. This paper investigates the following fundamental problem: how to judiciously plan the physical and the logical topologies of a WSN so that performance demands including network connectivity, sensing coverage quality, reliability, and lifetime are all satisfied with the least possible cost. To handle the uncertainty related to sensor connectivity and coverage, we devise a probabilistic-based communication cost model, and we exploit the belief functions theory to define a generic evidence fusion scheme that captures several characteristics of real-world applications. The uncertainty-aware cluster-based WSNs deployment problem is formulated as a multi-objective binary nonlinear and non-convex optimization problem, and an efficient heuristic using genetic algorithms is investigated. Using both simulations and testbed-based experiments, we show that the proposed deployment approach can fulfill the user performance needs, which confirms that the deployment of real-world fusion-based WSNs with predictable performance is possible.

Index Terms

Belief Functions Theory, Wireless Sensor Networks, Clustering, Fusion, Deployment, Coverage, Connectivity, Detection, Surveillance Applications, False Alarm.

I. Introduction

Effectively managing Wireless Sensor Networks (WSNs) is a major challenge starting from the deployment phase and throughout the entire lifetime of the network. Once a WSN is deployed, many optimization steps could be performed to enhance its performance. Therefore, many researchers have focused on approaches that improve the initial *physical topology* of the network such as the relocation of sensors [1].

An alternative approach is to focus on the *logical topology* of the WSN, which is formed by its communication graph. Hierarchical (tiered) architectures and hierarchical routing protocols [2] are important techniques in this area of research. In fact, picking the appropriate logical topology helps to reduce the radio interference, the probability of losses of messages, and the waiting time before data transmission. In addition, it eases data aggregation, which greatly decreases energy consumption, resulting in a prolonged network lifespan. Nevertheless, only a low

improvement could be expected with this approach, as the logical topology is mainly constrained by the physical topology of the network. Moreover, only a few metrics could be impacted by this approach. For instance, modifying the logical topology has little to no impact on coverage as the latter is more related to the sensors' positions.

As explained previously, many optimizations can be performed to enhance the performance of WSNs. Still, we consider that the most significant optimizations are those accomplished at the pre-deployment step to build the best possible topology that meets explicit performance requirements. In fact, the types, numbers, and locations of sensors have to be duly planned in order that performance requirements in terms of coverage, connectivity, and lifespan are all met while keeping the cost affordable. In the literature, this kind of deployment practice is referred to as *deterministic WSNs deployment*. There has been much related research on the deterministic deployment of WSNs in security monitoring [3], [4], [5]. These previous works are mainly based on a flat architecture. They focus solely on the coverage rate (characterized by the detection probability) and do not consider the quality of coverage (*i.e.,* they ignore false alarms).

Recently, Tan *et al.* [6] showed that the quality of coverage is greatly enhanced by data fusion when utilizing the collaboration among sensors. This has incited the development of new proposals [7], [8], [9], [10] that adopt data fusion techniques. These proposals consider tiered architecture and use different tools such as optimal control theory [7], [8], probability theory [9], [11], fuzzy theory [12], and Dempster-Shafer evidence theory [10]. Most of these approaches enhance the performance of the WSN by taking advantage of the collaboration between sensors. Nevertheless, these approaches do not take into account several deployment-related issues such as connectivity and network lifespan. Furthermore, most of them do not specify how and where the data fusion process should be handled, and how sensor data should be communicated to the fusion center. Therefore, practical situations are not handled by this type of research. Moreover, other approaches [11], [12] consider the base station as the fusion center. It is evident that this latter solution is not sustainable and could exhibit scalability issues.

In this work, we are attempting to bridge this gap by exploring simultaneously a cluster-based WSN architecture and an efficient Dempster-Shafer based fusion scheme while considering a full set of deployment-related issues, which is more suitable for surveillance applications. To the best of our knowledge, this paper is the first to comprehensively addresses the problem of fusion-based deterministic WSNs deployment by jointly considering sensing coverage quality, network connectivity, lifetime, reliability, and deployment cost. Consequently, the WSNs deployment issue becomes more complex but has more practical significance. The key contributions of this work are as follows.

1) The problem at hand is formulated as a nonlinear optimization problem with binary variables. Various aspects of real-life applications including uncertain sensor measurements, sensor's spatial distribution, unreliable connectivity, sensor reliability, and harsh deployment environments are captured in this framework.
2) As the formalized problem is NP-hard, an efficient heuristic is developed to solve it.
3) Using both simulations and testbed-based experiments, we show that the proposed deployment approach can fulfill the user performance needs.

The rest of this paper is structured as follows. Section II reviews related work. Section III summarizes the key

concepts of the evidence fusion model and presents our communication cost prediction approach. The uncertainty-aware cluster-based WSNs deployment problem is formalized in Section IV, and resolved in Section V. The evaluation of our proposed approach via simulations and experiments is presented in Section VI. Finally, Section VII draws the main conclusions of this paper.

II. Related work

The problem of cluster-based WSNs deployment could be approached in two different ways (levels), namely: at the logical topology or rather at the physical topology. In this section, previous works related to both approaches are discussed.

A. Clustering approaches

Clustering approaches [2], [13] target the logical topology of the network. More precisely, they target the routing layer. The main idea of these approaches is to group the sensors into clusters to achieve network scalability. Additionally, clustering has numerous advantages. It can conserve communication bandwidth, and reduce energy dissipation by using fusion schemes to minimize the number of messages forwarded to the sink. However, clustering approaches care mostly about network connectivity and route stability, without much concern about critical design goals of WSNs such as coverage [2], [14]. For these reasons, in the remainder of this paper, we will not discuss such approaches.

B. Deterministic sensor placement approaches

We refer to deterministic deployment strategies that exploit the clustering principle as *cluster-based deterministic deployment strategies*. There has been little research in this area. Previous deployment approaches have adopted simplistic combination techniques such as averaging. Recently, Xing *et al.* [15] showed by means of probability theory that value fusion enhances the coverage of WSNs.

Chang *et al.* [9] considered both detection probabilities and false alarm rates under a value fusion scheme [6], and formulated the WSNs deployment problem as a nonlinear and non-convex optimization problem. To solve this latter problem, they suggested the application of the constrained simulated annealing, which is a discrete global minimization algorithm of a high computational complexity. This limits their approach to rather small networks. To overcome this limitation, they devised a relatively low computational complexity heuristic. Simulation results were provided for the validation of the proposed method. Using the above-mentioned value fusion approach [6], Zhang *et al.* [16] presented a scheme to achieve barrier coverage in a hybrid wireless sensor network. The worst-case complexity of the proposed scheme is $\mathcal{O}(n^4)$ where n is the number of sensors.

Ababnah *et al.* [17] used the optimal control theory to formulate the deterministic deployment of WSNs as an optimal control problem, with the sensors' positions acting as control variables. In [7], this nonlinear optimal control problem was linearized and solved under several approximations while considering a value fusion scheme. The time complexity of the devised solution is $\mathcal{O}(3n^6 + 4n^4)$. In another recent work, a similar algorithm was introduced by

Ababnah *et al.* [8] where the decision fusion rule was implemented via the majority rule. The time complexity of the suggested algorithm is $\mathcal{O}(3n^6 + 4n^4)$. Table I presents a comparative overview of the above-discussed cluster-based deterministic deployment approaches.

TABLE I

A COMPARISON AMONG CLUSTER-BASED DETERMINISTIC DEPLOYMENT STRATEGIES.

Ref.	Fusion scheme	Primary objective	Secondary objective	Constraint	Complexity
[7]	Value fusion	Maximize detection performance	-	Fixed number of sensors	$\mathcal{O}(3n^6 + 4n^4)$
[8]	Decision fusion	Maximize detection performance	-	Fixed number of sensors	$\mathcal{O}(3n^6 + 4n^4)$
[9]	Value fusion	Maximize detection performance	Minimize the number of sensors	-	$\mathcal{O}(e^{n^2})$
[9]	Value fusion	Maximize detection performance	Minimize the number of sensors	-	$\mathcal{O}(n^2 m e^m)^*$
[16]	Value fusion	Strong barrier coverage	Minimize the number of active sensors	Fixed number of static and mobile sensors	$\mathcal{O}(n^4)$

* m is the mean number of spots within the impact region.

It is worth noting that previous works [7], [8], [9] rely on simplistic fusion approaches, and disregard completely several issues and challenges involved in the design of WSNs such as harsh deployment environments and sensor reliability. Very recently, in [10], we have considered a radically different approach by exploiting the belief functions theory [18] that allows taking into consideration not only the uncertainty in sensor data but also many aspects of real-life applications. More precisely, we have defined an evidence-based sensing mode where the sensor's output is modeled by a belief function rather than a Bayesian probability distribution. In addition, we have described the process of constructing belief functions from raw data returned by sensors [10]. This belief functions theory based fusion scheme has been exploited to devise a fusion-based deterministic deployment approach. Obtained results showed that significant improvement could be achieved when exploiting evidence combination.

The major drawbacks of all above-discussed fusion-based deterministic deployment approaches [7], [8], [9], [10] are:

- Although the generated topologies are hierarchical, the exact location of the cluster-heads is not determined. Indeed, these approaches focus on generating clusters, where each cluster comprises one cluster-head. However, they do not explicitly designate the cluster-heads within the clusters.
- Although sensors are supposed to collaborate, these approaches completely ignore network connectivity.
- The network lifespan issue is not addressed.

In the present paper, we tackle all of the above-mentioned drawbacks while investigating different mathematical tools. First, to handle the uncertainty in sensor data (caused by measurement errors due to the hardware or the environment), we exploit a belief functions theory based fusion scheme. Second, to take into consideration both network connectivity and lifespan, we devise a probabilistic-based model that allows an accurate prediction of communication cost and network lifespan. The next section describes these models in more detail.

III. COVERAGE, CONNECTIVITY, AND LIFESPAN MODELS

In this section, we present the mathematical models considered in this work. First, we summarize the key concepts of the evidence-based sensing model that we have proposed very recently in [10]. Second, we present our probabilistic-based communication cost model.

A. Coverage model

Unlike prior efforts, we consider an evidence-based decision scheme. In this latter, sensor nodes do not generate binary decisions but produce belief functions instead. In this section, we detail our method for constructing and manipulating these belief functions also referred to as basic belief masses (*bbm*).

1) Evidence construction: In surveillance applications, the main aim is to detect whether a target/event is present or not. The process of detection depends not only on the actual presence of the target/event but also on the RoI characteristics and the sensor's capabilities. Therefore, when a target/event is present in the RoI, the WSN will either correctly detect it or will miss it. From a modeling perspective, two states are necessary to express this fact: θ_0 (target/event present and not detected) and θ_1 (target/event present and detected). Consequently, our Frame of Discernment (FoD) is the set $\Theta^t = \{\theta_0, \theta_1\}$.

The uncertain raw sensory data can be translated into basic belief masses as follows: for a sensor s_i and relatively to a target/event at point $p \in RoI$, three inputs are defined: the mass assigned to no detection $m^{\Theta^t}_{d_{i/p}}(\theta_0)$, the mass assigned to detection $m^{\Theta^t}_{d_{i/p}}(\theta_1)$, and the unassigned mass $m^{\Theta^t}_{d_{i/p}}(\theta_0, \theta_1)$. By definition, the summation of the three belief masses always equals one. Therefore, although three masses are considered, there are only two independent ones. For instance, we can choose $m^{\Theta^t}_{d_{i/p}}(\theta_0)$ and $m^{\Theta^t}_{d_{i/p}}(\theta_1)$. The mass $m^{\Theta^t}_{d_{i/p}}(\theta_1)$ reflects the belief in a target/event presence at a location p, the *bbm* $m^{\Theta^t}_{d_{i/p}}(\theta_0)$ represents the opposite, and the *bbm* $m^{\Theta^t}_{d_{i/p}}(\theta_0, \theta_1)$ quantifies the sensor uncertainty.

Given a sensor s_i that believes, with a degree of belief $b_{d_{i/p}}$, that there is a target/event located at p, the following mapping is used to compute the probability masses:

$$m^{\Theta^t}_{d_{i/p}}(x) = \begin{cases} (1-u_i)\cdot(1-b_{d_{i/p}}) & \text{if } x = \{\theta_0\}, \\ (1-u_i)\cdot b_{d_{i/p}} & \text{if } x = \{\theta_1\}, \\ u_i & \text{if } x = \{\theta_0, \theta_1\}, \\ 0 & \text{if } x = \{\emptyset\}. \end{cases}$$

where $u_i \in [0, 1]$ is the uncertainty associated with the sensor s_i decision.

As explained above, when a target/event is actually not present in the RoI, the WSN could report a false detection. Such a decision is called a false alarm. From a modeling perspective, two states are necessary to express this fact: θ_2 (target/event absent and not detected), θ_3 (target/event absent and detected). Therefore, the FoD is the set $\Theta^{nt} = \{\theta_2, \theta_3\}$. Assuming that a sensor s_i has a false alarm rate b_{f_i}, its uncertain sensory data are translated into

basic belief masses by defining a belief function $m_f^{\Theta^{nt}}$ on Θ^{nt} as follows:

$$m_{f_i}^{\Theta^{nt}}(x) = \begin{cases} (1-u_i)\cdot(1-b_{f_i}) & \text{if } x = \{\theta_2\}, \\ (1-u_i)\cdot b_{f_i} & \text{if } x = \{\theta_3\}, \\ u_i & \text{if } x = \{\theta_2, \theta_3\}, \\ 0 & \text{if } x = \{\emptyset\}. \end{cases}$$

It should be noted here, that both formulations $m_{d_{i/p}}^{\Theta^t}$ and $m_{f_i}^{\Theta^{nt}}$ meet Appriou's axioms [19].

2) Decision combination: Decisions collected from multiple sensors are combined through the combination operation for final decision making. To combine distinct pieces of evidence on the same FoD, different combination rules could be used such as the Dempster-Shafer's rule of combination [18], Yager's rule [20], and the Proportional Conflict Redistribution rules [21]. In this work, we consider the Dempster-Shafer's rule of combination. For N sensors, the combination of the *N* belief functions $m_{d_1}^{\Theta^t}, \ldots, m_{d_N}^{\Theta^t}$ using the Dempster-Shafer's rule of combination yields the following belief function $m_d^{\Theta^t}$:

$$\begin{aligned} m_d^{\Theta^t}(\theta_0) &= 1 - \textstyle\prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_0)) \cdot k_d \\ m_d^{\Theta^t}(\theta_1) &= 1 - \textstyle\prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_1)) \cdot k_d \\ m_d^{\Theta^t}(\Theta^t) &= \textstyle\prod_{i=1}^{N} m_{d_i}^{\Theta^t}(\Theta^t) \cdot k_d \end{aligned} \tag{1}$$

where

$$k_d = \frac{1}{\prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_0)) + \prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_1)) - \prod_{i=1}^{N} m_{d_i}^{\Theta^t}(\Theta^t)}$$

In the same way, the combination of the *N* belief functions $m_{f_1}^{\Theta^{nt}}, \ldots, m_{f_N}^{\Theta^{nt}}$ using the Dempster-Shafer's rule of combination yields a *bba* $m_f^{\Theta^{nt}}$ that has the following expression:

$$\begin{aligned} m_f^{\Theta^{nt}}(\theta_2) &= 1 - \textstyle\prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_2)) \cdot k_f \\ m_f^{\Theta^{nt}}(\theta_3) &= 1 - \textstyle\prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_3)) \cdot k_f \\ m_f^{\Theta^{nt}}(\Theta^{nt}) &= \textstyle\prod_{i=1}^{N} m_{f_i}^{\Theta^{nt}}(\Theta^{nt}) \cdot k_f \end{aligned} \tag{2}$$

where

$$k_f = \frac{1}{\prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_2)) + \prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_3)) - \prod_{i=1}^{N} m_{f_i}^{\Theta^{nt}}(\Theta^{nt})}$$

3) Decision making: When decisions must be taken, the belief function undergoes a transformation called the pignistic transformation. This latter generates a probability distribution also known as the pignistic probability function. In this work, two pignistic transformations (referred to as $BetP_{d/p}$ and $BetP_{f/p}$) are constructed, relatively to a location $p \in RoI$, to allow the creation of the probabilities required for decision making. The pignistic transformation ($BetP_{d/p}$) of the belief function $m_d^{\Theta^t}$ is represented as follows:

$$\begin{aligned} BetP_{d/p}^{\Theta^t}(\theta_0) &= m_d^{\Theta^t}(\theta_0) + \tfrac{1}{2} m_d^{\Theta^t}(\theta_0, \theta_1) \\ BetP_{d/p}^{\Theta^t}(\theta_1) &= m_d^{\Theta^t}(\theta_1) + \tfrac{1}{2} m_d^{\Theta^t}(\theta_0, \theta_1) \end{aligned} \tag{3}$$

Likewise, the pignistic transformation ($BetP_{f/p}$) of the *bba* $m_f^{\Theta^{nt}}$ is represented as follows:

$$\begin{aligned} BetP_{f/p}^{\Theta^{nt}}(\theta_2) &= m_f^{\Theta^{nt}}(\theta_2) + \tfrac{1}{2} m_f^{\Theta^{nt}}(\theta_2, \theta_3) \\ BetP_{f/p}^{\Theta^{nt}}(\theta_3) &= m_f^{\Theta^{nt}}(\theta_3) + \tfrac{1}{2} m_f^{\Theta^{nt}}(\theta_2, \theta_3) \end{aligned} \tag{4}$$

We define the concept of (α,β)-coverage that allows quantifying the quality of coverage delivered by a WSN as follows.

Definition 1. *Let $\alpha \in [0,1]$ and $\beta \in [0,1]$ be two constants, a location $p \in RoI$ is (α,β)-covered if:*

$$\begin{aligned} BetP_{d/p}^{\Theta^{t}}(\theta_1) &\geq \alpha \\ BetP_{f/p}^{\Theta^{nt}}(\theta_3) &\leq \beta \end{aligned} \tag{5}$$

In [10], we have demonstrated that the false alarms rate generated by a WSN could be minimized or lessened to zero by assuming the above-described evidence-based fusion approach. We have also shown how this latter can be straightforwardly expanded to handle deployment-related problems such as harsh deployment environments and sensor reliability.

B. Probabilistic-based communication cost model

It is customary to assume a binary connectivity model according to which two sensors are able to communicate directly if the Euclidean distance separating them is less than or equal to a communication range R_c [22]. This binary model simplifies the analysis by approaching the network connectivity problem from a geometric perspective, it remains however limited and not realistic. In fact, empirical research [23] demonstrate that there are no clear-cut boundaries between successful and failed communications.

In practice, the received wireless signal strength is susceptible to noise, interference, and other wireless radio channel impediments. Usually, the variation in received signal strength is characterized in terms of path loss and shadowing. The former describes the amount of wireless signal loss (attenuation) between a receiver and a transmitter whereas the latter is caused by obstacles between the transmitter and receiver that attenuate signal strength through absorption, reflection, scattering, and diffraction.

Empirical measurements have demonstrated that shadowing could be approximated by a zero-mean normal distribution with a standard deviation of σ_ϵ. As each environment has unique characteristics, most radio wave propagation models adopt a mixture of empirical and analytical approaches. For a wide range of environments, the log-normal shadowing path loss model [24] is one of the most used radio wave propagation models. It is given by:

$$PL(d) = PL(d_0) + 10\gamma \log_{10}(\frac{d}{d_0}) + \epsilon \tag{6}$$

where $PL(d_0)$ is the path loss (in dB) at a reference distance d_0, $PL(d)$ is the path loss (in dB) at an arbitrary distance $d > d_0$, γ is the path loss exponent (could varies from 2 in free space to 6 inside buildings), and ϵ is a zero-mean Gaussian distributed random variable (in dB) with a standard deviation σ_ϵ. This latter is used only when there is a shadowing effect; otherwise, it is zero.

The received signal strength (P_r) at a distance d is usually represented with the following formula:

$$\begin{aligned} P_r(d) &= P_t - PL(d) \\ &= P_t - PL(d_0) - 10\gamma \log_{10}(\tfrac{d}{d_0}) - \epsilon \end{aligned} \tag{7}$$

Fig. 1 illustrates an analytical propagation model for $\gamma = 2$, $\sigma_\epsilon = 4$, $PL(d_0) = 55dB$, $d_0 = 1$ and an output power $P_t = 0dBm$ (*e.g.,* CC2420).

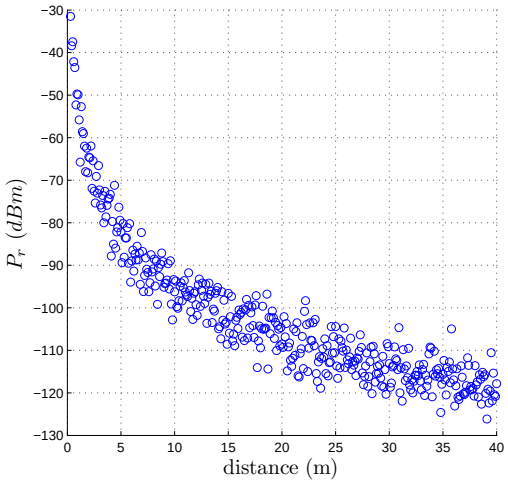


Fig. 1. Channel Model, $\gamma = 2$, $\sigma_\epsilon = 4$, $P_t = 0dBm$.

According to Equation 7, $P_r(d)$ is Gaussian where $P_r(d) \sim \mathcal{N}(P_t - PL(d_0) - 10\gamma \log_{10}(\frac{d}{d_0}), \sigma_\epsilon)$. Considering two sensors s_i and s_j located at distance d from each other, the probability of a successful transmission between s_i and s_j can be easily calculated as follows:

$$\mathcal{P}[P_r(d) > SS_{min}] = Q\left(\frac{SS_{min} - (P_t - PL(d_0) - 10\gamma \log_{10}(\frac{d}{d_0}))}{\sigma_\epsilon}\right) \tag{8}$$

A transmission is considered successful if its received signal strength is greater than a certain threshold SS_{min} which is the minimum acceptable signal strength. $Q(.)$ denotes the complementary cumulative distribution function (CCDF) of a standard Gaussian random variable:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{\frac{-t^2}{2}} \mathrm{d}t$$

Hence, for a successful communication between s_i and s_j, the expected number of retransmissions nRT is:

$$nRT(d) = \frac{1}{\mathcal{P}[P_r(d) > SS_{min}]} \tag{9}$$

Fig. 2 illustrates the connectivity model formulated above. It clearly shows that in the connectivity range, the received signal strength by some areas is lower than SS_{min}. Whereas, outside the connectivity range, the received signal strength by some area is higher than SS_{min}. As stated before, this model captures the fact that there are no clear-cut boundaries between successful and failed communications.

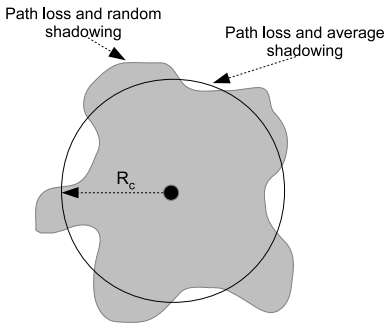


Fig. 2. 2D visualization of the connectivity model.

On the basis of the connectivity model described above, we can define the communication cost of a link and by extension that of a cluster as follows.

Definition 2. *The expected number of retransmissions (denoted nRT) required for a successful transmission between two nodes s_i and s_j represents the communication cost of the link ij (hereafter denoted Lc_{ij}), given by:*

$$Lc_{ij} = nRT(||ij||) = \frac{1}{\mathcal{P}[P_r(||ij||) > SS_{min}]}$$

Definition 3. *The summation of all communication cost of the links connecting the cluster members to the cluster-head (denoted h) represents the cluster's communication cost (hereafter denoted Cc), given by:*

$$Cc_h = \sum_{s_i \in \text{ cluster members}} Lc_{ih}$$

IV. Problem formalization

This section formalizes the uncertainty-aware cluster-based WSNs deployment problem while considering the evidence-based sensing model and the communication cost model, both described in the previous section.

A. Network model and assumptions

The set $\mathbb{T} \subseteq RoI$ is defined by the locations of the targets/events that appear in the RoI. These known locations are referred to as *target points*. $\mathbb{T}$ is determined based on the user needs. For instance, target points could be hand-picked if a priori information about the possible locations of targets/events is available. Otherwise, target points could be uniformly scattered over the RoI.

The set $\mathbb{D} \subseteq RoI$ is defined by the locations that could host the sensors. These locations are referred to as *deployment points*. We assume that, for each target point $t \in \mathbb{T}$, only a subset of deployed sensors (hereafter denoted $\mathbb{N}(t)$) participate in the fusion process. $\mathbb{N}(t) \subseteq \mathbb{D}$ is a design parameter, which is determined by the user. As will be discussed in more detail in Section VI, $\mathbb{N}(t)$ is mainly constrained by the communication overhead.

We also assume two user-defined inputs α_p and β_p associated with each target point $p \in \mathbb{T}$. α_p is the minimum target/event detection probability threshold, whereas β_p is the maximum false alarm rate threshold. In the case of a uniform (α,β)-coverage, only two values α and β are defined for the whole set $\mathbb{T}$.

B. Problem formulation

The uncertainty-aware cluster-based WSNs deployment problem is defined as the problem of determining the minimum number of sensors and their locations such that the set of target points $\mathbb{T}$ is (α,β)-covered while considering the evidence-based sensing model presented in Section III-A. Deployment-related issues such as network lifespan and connectivity may be included by considering the probabilistic-based communication cost model defined in Section III-B. In this work, besides the sensing quality, we consider the communication cost and lifespan objectives. Thus, we have to simultaneously optimize sensing quality, communication cost, and network lifespan.

Since transmission consumes more energy than any other process, we have to ensure that our sensor placement has reliable links and the overall number of unnecessary retransmissions is minimized. For that, we have to guarantee that the formed clusters minimize the communication cost as defined in Section III-B. These objectives are competing with each other, and therefore, we have Pareto-optimal solutions. For instance, the (α,β)-coverage objective will tend to scatter the network topology to minimize the sensing overlap, whereas the communication cost objective will favor topologies where sensors are not located too far from each other to minimize communication cost and enhance the network lifespan.

Our aim is to generate a good cluster topology where each sensor can transmit directly to the cluster-head. It should be noted here that the expected communication activities carried out by the cluster-heads would lead to excessive energy consumption. To deal with this heavy energy burden of cluster-heads at least two solutions are likely. The first solution is to deploy cluster-heads that are more energy-rich as compared to the sensor nodes. As the positions occupied by cluster-heads are known, the second solution is to redeploy a cluster-head once its energy level reaches a minimum threshold.

The generated topology should ensure full (α,β)-coverage and satisfies the network connectivity constraint while consuming less energy. Additionally, the sensors in the same cluster should be highly connected, and less connected to the sensors in other clusters. There are other problems to be tackled when using clustering methods, such as the number of orphans (isolated) sensors, the number of orphans cluster-heads, and the cluster size, to name few. Consequently, besides the sensing quality objective, we look for the optimal network topology that best:

- Reduces cluster-heads number;
- Reduces the orphans sensors number (guarantees that each sensor has at least one cluster-head in its vicinity);
- Reduces the orphans cluster-heads number (guarantees that each cluster-head has some sensors in its vicinity);
- Reduces the overlapping areas of clusters (strives to distribute or select the cluster-heads in such a way that there is a minimum overlapping).
- Reduces unnecessary retransmissions overall number.

Formally, the problem at hand is as follows:

$$
\begin{aligned}
&\min \quad \sum_{p \in \mathbb{D}} x_p \\
&\min \quad \sum_{p \in \text{ cluster-heads}} x_p \\
&\min \quad \sum_{p \in \text{ isolated sensors}} x_p \\
&\min \quad \sum_{p \in \text{ isolated cluster-heads}} x_p \\
&\min \quad \sum_{\text{cluster-heads}} \text{cluster-heads overlapping} \\
&\min \quad \max_{h \in \text{ cluster-heads}} (Cc_h) \\
&\text{s.t.} \quad BetP^{\Theta^t}_{d/p}(\theta_1) \geq \alpha_p \qquad , \forall p \in \mathbb{T} \\
&\qquad\quad BetP^{\Theta^{nt}}_{f/p}(\theta_3) \leq \beta_p \qquad , \forall p \in \mathbb{T} \\
&\qquad\quad x_p \in \{0,1\} \qquad , \forall p \in \mathbb{D}
\end{aligned}
$$

This formulation improves the cluster connectivity as well as the cluster lifetime, and therefore, the network lifetime. The position of the sink(s) could be easily considered in this formulation. For instance, the straightforward approach is to minimize the distances between the cluster-heads and the sink(s). It is interesting to note that the issue at hand is a binary nonlinear and non-convex multi-objective optimization problem. In the following, we investigate a multi-objective optimization technique to find the global Pareto-optimal solutions.

V. Sensor placement algorithm

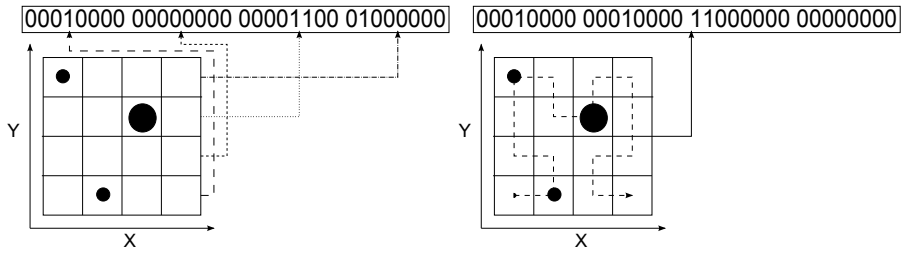
Many conflicting objectives typify the nonlinear and non-convex optimization problem described in the previous section. Consequently, it is far essential to deal with that problem as a multi-objective optimization problem. In such a problem, there is usually no single optimal solution, but rather a set of alternative solutions known as the Pareto-optimal solution set, which is the set of solutions such that attempting to improve any objective function would necessarily worsen the other objective values. Pareto-optimal solutions could be obtained for the problem at hand by using multi-objective optimization methods.

Classical optimization methods are limited in handling multi-objective optimization problems, as they transform the multi-objective optimization problem into a single objective optimization problem to find only a single Pareto-optimal solution. In this work, we investigate the multi-objective genetic algorithm (GAs). The advantage of this latter is that it can find the whole Pareto-optimal solutions in a single run [25]. A number of multi-objective GAs were developed in the last few years. Among the existing GAs for effectively dealing with multi-objective optimization problems, the most prominent ones are the Pareto archived evolutionary strategy (PAES) [26], the Strength Pareto evolutionary algorithm (SPEA-2) [27], and the Non-dominated sorting genetic algorithm (NSGA-II) [28]. It has been shown that NSGA-II can converge to the true Pareto-optimal front while ensuring a good

distribution of Pareto-optimal solutions [28]. Furthermore, the complexity of NSGA-II is lower than that of other multi-objective GAs. For that, on the basis of NSGA-II, we devise a new algorithm to solve our multi-objective optimization problem. In the remainder of this paper, this algorithm will be referred to as *MGEBDA* (Multi-Objective Genetic Evidence-Based Deployment Algorithm).

A. Problem Representation

In MGEBDA, the number and locations of sensors (members and cluster-heads) are specified by an individual which is a candidate sensor placement. This latter is simply modeled by a bit string of length $L = 2|\mathbb{D}|$ where each two bits represent a deployment point $p \in \mathbb{D}$ with the following meaning: (i) "00" no sensor to be deployed, (ii) "01" or "10" a regular sensor to be deployed, and (iii) "11" a cluster-head to be deployed. To exploit this linear representation, a transformation from 2D (or even 3D) to 1D space is necessary. For instance, the linear horizontal (or vertical) coding (Fig. 3(a)) is a straightforward transformation. Other transformations such as the Z-order curve and the Hilbert space-filling curve (Fig. 3(b)) could also be applied as they preserve fairly well locality. Fig. 3 shows the binary representation of a placement of two sensors and one cluster-head within a 2D 4×4 RoI (here $\mathbb{D} = RoI$) using two transformations.



(a) Linear horizontal mapping. (b) Hilbert curve mapping.

Fig. 3. Example of a sensor placement representation.

B. Constraints Handling

In this work, we adopt the idea of the constrained-domination principle [28] to handle constraints. The idea of this approach is based on a simple modification of the definition of the *domination* relationship between any two solutions x_p and x_q.

Definition 4. *Given two solutions x_p and x_q, x_p is said to* constrain-dominate *x_q, if any of the following conditions is true:*

(i) *x_p is feasible and x_q is not;*

(ii) *x_p and x_q are both infeasible, but x_p has a smaller overall constraint violation;*

(iii) *x_p and x_q are both feasible and x_p dominates x_q.*

The previously-defined constrained-domination principle ensures that any infeasible solution has a smaller non-domination rank than all feasible solutions, which are ranked based on their level of non-domination according to the values of the objective function [28]. However, between two infeasible solutions, the solution with a lower constraint violation is favored.

C. Genetic Operators

As in NSGA-II, MGEBDA uses the bitwise mutation with a small probability p_m. However, unlike NSGA-II that uses the one-point crossover, MGEBDA combines three crossover operators: uniform, two-point, and one-point crossover (Fig. 4). The frequency with which MGEBDA applies the crossover operators is controlled by the crossover rate p_c.

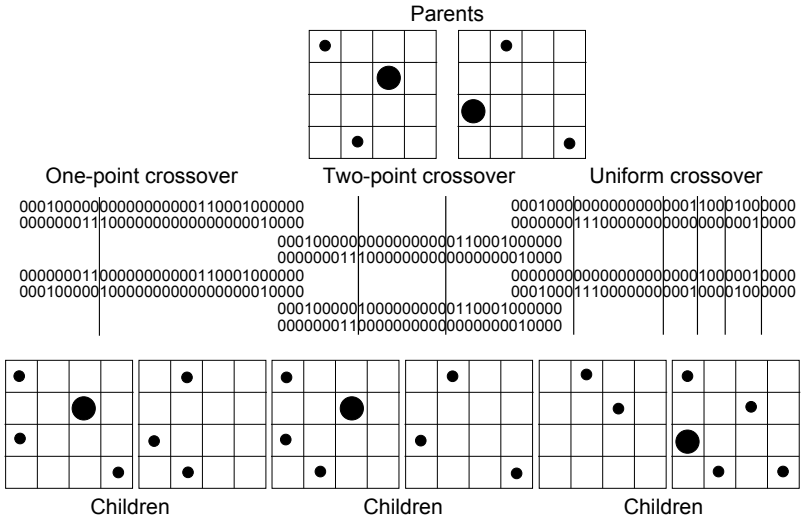


Fig. 4. Crossover operators used in MGEBDA.

D. Evolution Engine

MGEBDA uses the evolution engine of NSGA-II. Selection can be done based on the crowding distance operator as follows: a sensor placement x_p wins the tournament with another sensor placement x_q if:

(i) x_p has higher rank than x_q, or

(ii) x_p and x_q have the same rank, but x_p has a larger crowding distance than x_q.

Regarding the replacement strategy, the elitism is implemented by applying the constrained non-dominating sorting on the union of offspring and parent populations. As a result of this operation, the entire population will be sorted into fronts. The new population for the next generation is filled up with the lower ranking fronts solutions while favoring the ones with the largest crowding distance.

E. Description of MGEBDA

Algorithm 1 outlines the pseudo-code of the proposed sensor placement algorithm.

Algorithm 1 *MGEBDA*

Input: external parameters ($\mathbb{D}$, $\mathbb{T}$, coverage model, ...) and internal parameters (population size N, p_m, p_c, ...)

Output: Pareto-optimal solutions

1: Initialize the first population (of size N) using randomly generated sensor placements
2: **while** (stopping conditions are not met) **do**
3: Compute the objective values
4: Assign rank to sensor placements using the constrained non-dominating criteria ▷ *using Definition 4*
5: Compute the crowding distance of each sensor placement
6: Apply the evolution engine to select the sensor placements from the population
7: Produce new sensor placements by employing the genetic operators with their respective probabilities
8: Apply the evolution engine to create the next population
9: **end while**
10: Return the population's first front

VI. Performance Evaluation

The effectiveness and efficiency of the proposed deployment approach are evaluated by carrying out numerical simulations and testbed-based experiments.

A. Numerical experiments

In this section, we carry out numerical experiments to assess the performance of the proposed deployment approach. First, MGEBDA performances are analyzed for different criteria settings. Second, we evaluate the impact of the fusion radius on the performance of the proposed deployment approach. Finally, we assess the repercussion of the requested connectivity reliability on the deployment cost. MGEBDA implementation is based on the ParadisEO framework [29]. We consider that the deployment and target points are regularly spaced on a grid, therefore the RoI is a $k \times k$ grid. Unless otherwise specified, false alarm and detection requirements are assumed uniform over the RoI, with $\alpha = 0.9$ and $\beta = 0.01$. The simulations parameters are described in Table II.

To evaluate the MGEBDA performance or multi-objective algorithms in general, several metrics appeared in the literature [30]. However, most of the recent proposals [30] consider that the global Pareto front of the multi-objective optimization under study is known which is not our case. Furthermore, the few metrics that can be employed when considering the global Pareto front is unknown, such as the spread metrics, only judge the uniformity of solution points over different subregions without pointing out the overall quality of solutions.

To assess the performance of MGEBDA for different parameter settings and for the reasons discussed above, we propose to plot the Pareto fronts to enable an enhanced visual inspection including the locations of the fronts,

TABLE II
SIMULATIONS PARAMETERS

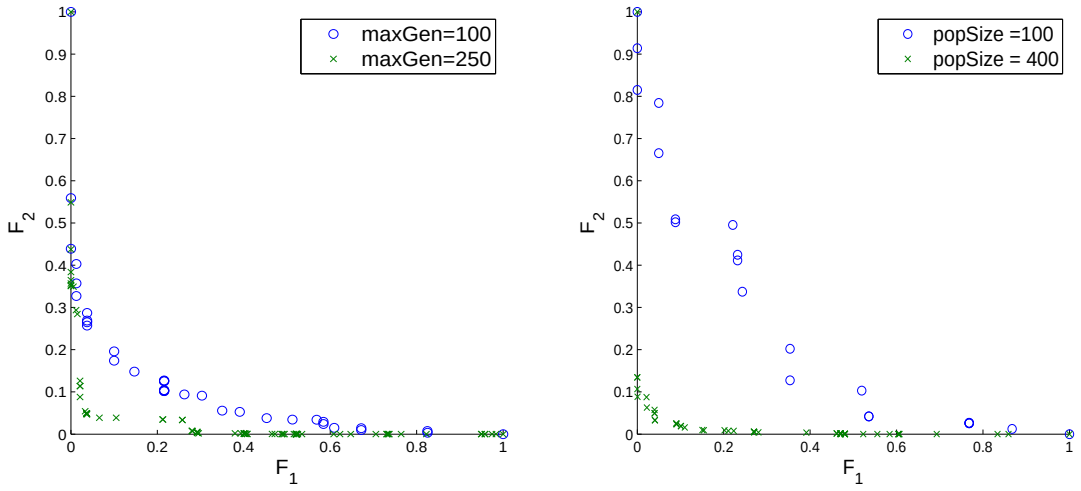
Parameters	Values
k	$10 - 100$
α, β	$0.9, 0.01$
Fusion radius	$2 - 9$
SS_{min}, γ, σ_ϵ	$-70dBm$, 2, $4dB$
$PL(d_0)$, d_0, P_t	$55dB$, $1m$, $0dBm$
$popSize$	100, 400
$maxGen$	100, 250

the number of points in each front, and their distribution. For illustration purpose, the objectives considered in the deployment problem were grouped into two objective functions F_1 and F_2. The F_1 function measures the deployment cost along with the degree of detection rate constraints dissatisfaction and the degree of false alarm rate constraints dissatisfaction. The F_2 function measures all other objectives. Both functions, F_1 and F_2, are to be minimized.

1) Impact of the key parameters: In this section, some experiments are performed to illustrate the impact of a couple of different criterion settings on the MGEBDA performance. Precisely, we analyze the effect of the population size and the number of maximum generations on the MGEBDA performance. For that, first all other criteria are fixed, but the number of maximum generations is increased from 100 to 250. Second, we increase the population size from 100 to 400 while keeping all other parameters unchanged.

Fig. 5 shows one of the ten runs of MGEBDA with different parameters settings. First, this figure shows the ability of MGEBDA to converge to the true Pareto front and to find various solutions in the front. Second, increasing the size of the population and/or the number of maximum generations help MGEBDA to converge very close to the true Pareto-optimal front and maximize the number of elements of the Pareto-optimal set found. For instance, with 100 and 250 generations, MGEBDA finds 35 and 81 Pareto-optimal solutions, respectively. A large population size and/or a large number of maximum generations will undoubtedly raise the convergence of MGEBDA. However, it will require more computations per generation which slows the execution of MGEBDA. It is worth mentioning that for large-scale scenarios, an estimation of the required number of sensors and cluster-heads could speed up the convergence time; this could be achieved by exploiting approaches such as [31]. We have also analyzed the effect of these parameters on various RoI. We found a population size of $2 \times L$ and a number of maximum generations of 500 to be a good trade-off.

2) Impact of fusion radius: As discussed previously, one of the most important design criteria of the fusion model is the user-defined fusion radius, which is essentially constrained by the communication overhead. In this section, various numerical results are given to provide a good understanding of the fusion radius impact.

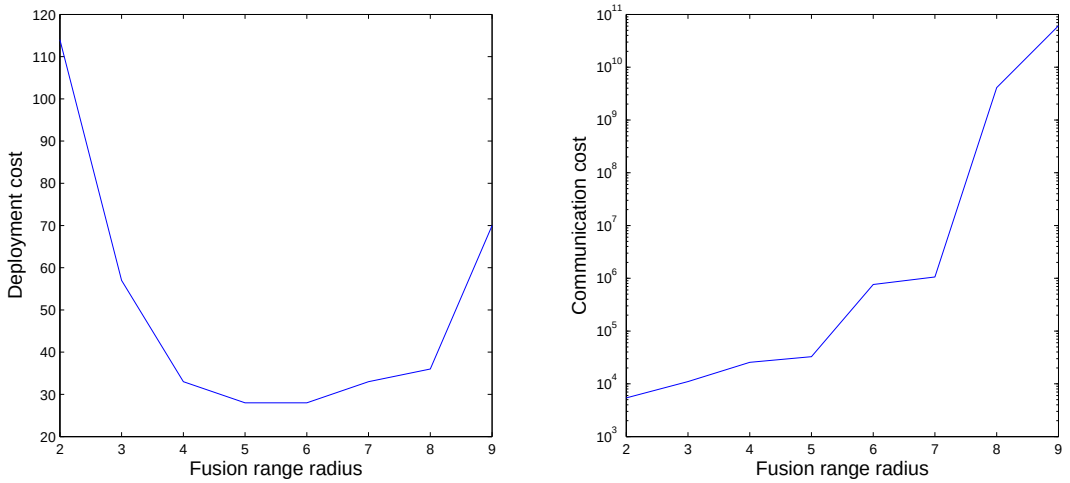


(a) Pareto frontier with different number of maximum generations.

(b) Pareto frontier with different population size.

Fig. 5. Pareto frontier for MGEBDA with different parameters settings.

To determine the impact of the fusion radius on the deployment and communications costs, the radius is varied from 2 to 9 in a 20×20 RoI. Fig. 6 shows, as a function of the fusion radius, the deployment and communication costs (as stated by Definitions 2 and 3) needed for full uniform $(0.9, 0.01)$-coverage. It is worth to note that as several Pareto-optimal solutions can be found by MGEBDA, each run the best solution is considered in terms of cost of deployment.

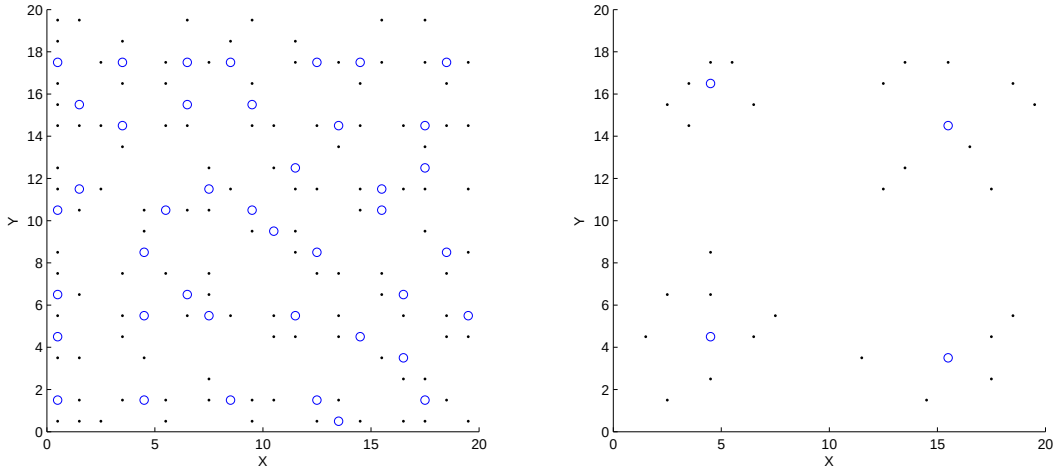


(a) Deployment cost.

(b) Communication cost.

Fig. 6. Costs *vs.* fusion radius.

Fig. 6 shows the impact of the fusion radius on the performance in terms of deployment and communication costs. Two observations are worth noting. First, when the fusion radius is increased from 2 to 6, the deployment cost rapidly dropped from 114 to 32, and then increases gradually to reach 70 when the fusion radius became larger. Second, the communication cost increases exponentially with the increase of the fusion radius from 2 to 9. This is due to the fact that with the increase of the fusion radius, more distant sensors pay off the data fusion, leading to an improved sensing quality and a lower deployment cost in terms of both sensors and cluster-heads. Fig. 7 illustrates the network topologies generated by MGEBDA while considering two fusion radii. As can be seen in this figure, a fusion scheme with a fusion radius of 6 uses fewer sensors and cluster-heads as compared to a fusion scheme with a fusion radius of 2.



(a) Fusion radius of 2.

(b) Fusion radius of 6.

Fig. 7. Generated topologies *vs.* fusion radius.

Nevertheless, the information quality of the sensors distant from the target point decreases with the increase of the fusion radius. Therefore, their fusion leads inevitably to inferior detection performance, which increases the deployment cost. Consequently, a longer distance between the sensors and their cluster-head results in a higher communication cost.

3) The effect of the requested connectivity reliability: Through this experiment, we analyze the impact of the requested connectivity reliability on the deployment cost (*i.e.,* the number of deployed sensors and cluster-heads). For that, in a 20×20 RoI, we vary the minimum acceptable signal strength (SS_{min}) and we analyze the deployment cost. Table III summarizes the obtained results.

We notice that increasing the requested connectivity reliability does not increase the number of regular sensors. However, the structure of the generated topology, in terms of clusters distribution, was highly affected. In fact, we can clearly see that increasing the requested connectivity reliability increases the number of cluster-heads and therefore increases the number of clusters. This can be explained by the fact that to ensure a high connectivity

TABLE III

THE EFFECT OF THE REQUESTED CONNECTIVITY RELIABILITY.

SS_{min} (dBm)	Number of sensors	Number of CHs
-90	28	5
-60	27	8
-50	27	9

reliability, the distances separating sensors members and their corresponding cluster-heads are required to be small in order to guarantee links with good quality which, inevitably, reduce the size of the clusters. Nevertheless, these good links help to reduce the communication cost of the network. To summarize, when increasing the requested connectivity reliability, MGEBDA tends to generate small clusters (in terms of distances separating sensor members and their corresponding cluster-heads).

B. Testbed-based experiments

To more investigate the actual benefit of our proposed deployment approach, we have developed a PIR based wireless sensor network testbed termed as *ArduiNet*. This motion detection testbed is based on the Arduino platform [32] which is an open-source electronics prototyping platform. In the sequel, the testbed architecture and its components are detailed, along with experimental evaluations.

1) Software, hardware, and system architecture: Fig. 8 shows an ordinary ArduiNet node. It comprises an Arduino UNO board, a PIR motion sensor, an XBee module (IEEE 802.15.4), and a Wireless Proto shield. Here, an analog PIR Phidgets 1111_0 sensor (Panasonic AMN23111) is connected to the Arduino Uno board. The PIR sensor detects sudden changes in the infrared landscape within its field of view. These changes are caused by the movement of a person (or object) whose temperature varies from that of the surroundings.

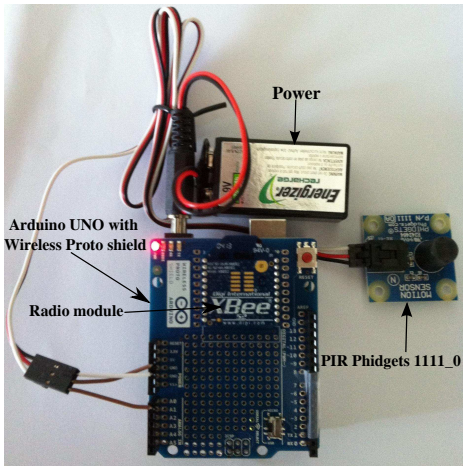


Fig. 8. An ordinary ArduiNet node.

A $2.4m \times 4m$ lab room was used to host our ArduiNet deployment. To do so, a two-dimensional metallic grid was installed on the roof in order to hold the ArduiNet nodes as can be seen in Fig. 9. The intersections of the grid lines are our deployment points (Fig. 9).

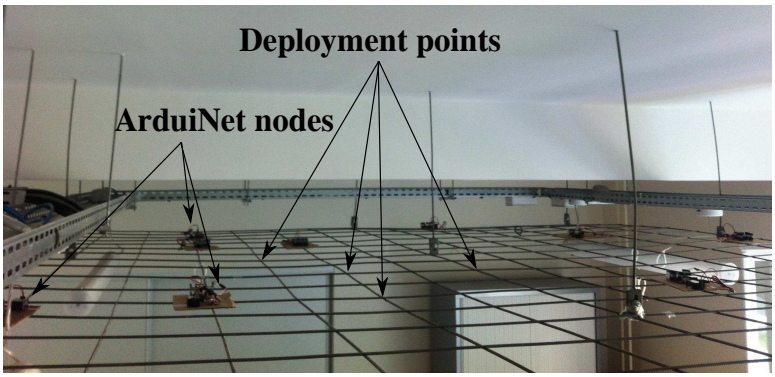


Fig. 9. System architecture.

In the sequel, we discuss the acquired results. We describe how to construct mass functions from the analog PIR Phidgets 1111_0 raw sensory data. Then, we evaluate the performance of ArduiNet as a fusion-based surveillance system enriched with the proposed approach.

2) Constructing mass functions: We employ the simple approach described in [10] to create belief functions from sensing data. Fig. 10 shows a set of belief functions for the considered PIR sensor performing in our lab room. This sensor returns a measure around 500 when there is no motion. Measures out of the 400 to 600 range correspond to motion detection with a high probability. Before the exploitation of ArduiNet, the set of belief functions is constructed empirically by observing the measures returned during several experiments. During the exploitation phase of ArduiNet, each time the cluster-head receives a measure, it performs a projection on the previously constructed set to acquire the corresponding mass function. For example, if the ArduiNet node returns a measure of 100 (see Fig. 10), then the resulting *bba* will have three focal sets: $m(\{\theta_0\}) = 0.05$, $m(\{\theta_1\}) = 0.94$, and $m(\Theta) = 0.01$.

3) The performance of ArduiNet: As explained previously, the ArduiNet is endowed with the proposed approach to build a fusion-based surveillance system. The experimental performances of this latter are discussed in this section.

The leading idea of the proposed approach is to deploy and use the WSN according to the physical and logical topologies that were established during the pre-deployment phase by the sensor placement algorithm fed with the design objectives. Actually, for different physical and logical topologies, the WSN performances are estimated by simulations before its deployment, and the best possible configuration that meets the user requirements is selected. After that, the WSN real-world deployment and usage must adhere to the selected topologies. For example, Fig. 11 shows a visualization of a sensor placement outputted by the sensor placement algorithm for the lab room scenario with (0.90, 0.01)-coverage.

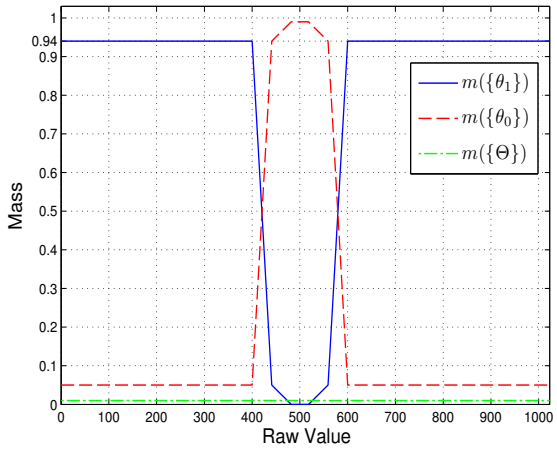


Fig. 10. Example of a set of mass functions related to the analog PIR sensor Phidgets 1111_0.

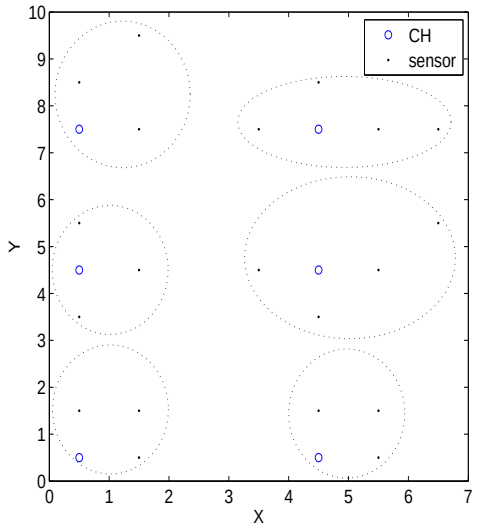


Fig. 11. A visualization of MGEBDA's output for the lab room scenario.

In the experimental tests, we deploy and use ArduiNet as defined by the MGEBDA's output. We allow a person to walk inside the testbed lab room through several paths at different speeds, and we measure the achieved (α,β)-coverage. As explained above, each cluster-head locally makes the detection decision based on the received measurements. The described experiments are repeated more than five thousand times. Table IV illustrates a summary of the obtained results.

The obtained results show that the achieved detection probability is greater than the requested ones, in both cases. Moreover, a zero false positive rate is acquired. These results are due to the fact that the devised evidence fusion scheme can remarkably improve (α, β)-coverage through the collaboration of nearby sensors. Furthermore, these

TABLE IV
EXPERIMENTAL RESULTS

Scenarios	Scenario #1	Scenario #2
Requested (α,β)-coverage	(0.90, 0.01)	(0.95, 0.01)
Deployment cost	26	29
Achieved (α,β)-coverage	(0.92, 0.00)	(0.96, 0.00)

experiments were accomplished in an indoor area, thus a very low false positive rate was expected. The results of these additional tests confirm the previously obtained results. The proposed deployment approach manages to achieve the requested user requirements in terms of both detection and false alarm rates.

VII. CONCLUSION

Deployment is of the utmost importance in the process of developing WSNs solutions for real-life applications as it decides the available resources and their configuration for system setup. This, in turn, plays a key role in network performance. In this paper, a robust uncertainty-aware cluster-based deterministic deployment approach has been presented. This work presents a contribution to the current literature by providing a comprehensive deployment approach that considers a set of factors involved in the deployment of WSNs. The considered factors reflect several characteristics of real-world applications such as uncertain sensor measurements, the spatial distribution of sensors, unreliable connectivity, harsh deployment environments, and sensor reliability. The described approach was evaluated by both simulation and testbed experiments. The obtained results show that the proposed approach facilitates the deployment of real-world fusion-based wireless sensor networks while achieving the expected network performances.

REFERENCES

[1] M. R. Senouci, M. Abdelhamid, and K. Assnoune, "Localized movement-assisted sensor deployment algorithm for hole detection and healing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1267–1277, 2014.

[2] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, pp. 2826–2841, 2007.

[3] M. P. Fanti, G. Faraut, J. J. Lesage, and M. Roccotelli, "An integrated framework for binary sensor placement and inhabitants location tracking," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, pp. 154–160, Jan 2018.

[4] B. Cao, J. Zhao, Z. Lv, and X. Liu, "3d terrain multiobjective deployment optimization of heterogeneous directional sensor networks in security monitoring," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2017.

[5] D. Thomas, R. Shankaran, M. Orgun, M. Hitchens, and W. Ni, "Energy-efficient military surveillance: Coverage meets connectivity," *IEEE Sensors Journal*, vol. 19, pp. 3902–3911, May 2019.

[6] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia, "Exploiting data fusion to improve the coverage of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 450–462, 2012.

[7] A. Ababnah and B. Natarajan, "Optimal sensor deployment for value-fusion based detection," in *GLOBECOM'09*, pp. 1–6, 2009.

[8] A. Ababnah and B. Natarajan, "Lqr formulation of sensor deployment for decision fusion based detection," in *IEEE GLOBECOM*, pp. 1–5, 2010.

[9] X. Chang, R. Tan, G. Xing, Z. Yuan, C. Lu, Y. Chen, and Y. Yang, "Sensor placement algorithms for fusion-based surveillance networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, pp. 1407–1414, August 2011.

[10] M. R. Senouci, A. Mellouk, N. Aitsaadi, and L. Oukhellou, "Fusion-based surveillance wsn deployment using dempster-shafer theory," *Journal of Network and Computer Applications*, vol. 64, pp. 154–166, April 2016.

[11] L. Yang, J. Liang, and W. Liu, "Graphical deployment strategies in radar sensor networks (rsn) for target detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, p. 55, Mar 2013.

[12] M. Collotta, G. Pau, and A. V. Bobovich, "A fuzzy data fusion solution to enhance the QoS and the energy consumption in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–10, 2017.

[13] F. Fanian and M. K. Rafsanjani, "Cluster-based routing protocols in wireless sensor networks: A survey based on methodology," *Journal of Network and Computer Applications*, May 2019.

[14] A. Ghosal and S. Halder, "Lifetime optimizing clustering structure using archimedes' spiral-based deployment in wsns," *IEEE Systems Journal*, vol. 11, pp. 1039–1048, June 2015.

[15] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C. Yi, "Data fusion improves the coverage of wireless sensor networks," in *MobiCom '09*, (New York, USA), pp. 157–168, 2009.

[16] X. Zhang, M. L. Wymore, and D. Qiao, "Cost-efficient barrier coverage with a hybrid sensor network under practical constraints," in *IEEE ICC*, (Paris, France), 2017.

[17] A. Ababnah, *Sensor deployment in detection networks - a control theoretic approach*. PhD thesis, Kansas State University, 2010.

[18] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[19] A. Appriou, "Formulation et traitement de l'incertain en analyse multisenseurs," in *GRETSI*, pp. 951–954, 1991.

[20] R. R. Yager, "On the dempster-shafer framework and new combination rules," *Inf. Sci.*, vol. 41, pp. 93–137, Mar. 1987.

[21] F. Smarandache and J. Dezert, *Advances and Applications of DSmT for Information Fusion (Collected Works)*. Rehoboth, NM, USA: American Research Press, 2004.

[22] M. R. Senouci and A. Mellouk, *Deploying Wireless Sensor Networks: Theory and Practice*. ISTE and Elsevier, March 2016.

[23] H. Nikookar and H. Hashemi, "Statistical modeling of signal amplitude fading of indoor radio propagation channels," in *Proceedings of 2nd IEEE International Conference on Universal Personal Communications*, (Ottawa, Ontario, Canada), IEEE, Oct. 1993.

[24] T. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2nd ed., 2001.

[25] E.-G. Talbi, *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.

[26] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation," in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, pp. 98–105, 1999.

[27] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - a comparative case study," in *Parallel Problem Solving from Nature* (A. Eiben, T. Bck, M. Schoenauer, and H.-P. Schwefel, eds.), vol. 1498 of *Lecture Notes in Computer Science*, pp. 292–301, Springer Berlin Heidelberg, 1998.

[28] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[29] A. Liefooghe, L. Jourdan, T. Legrand, J. Humeau, and E.-G. Talbi, "Paradiseo-moeo: A software framework for evolutionary multi-objective optimization," in *Advances in Multi-Objective Nature Inspired Computing* (C. Coello Coello, C. Dhaenens, and L. Jourdan, eds.), vol. 272 of *Studies in Computational Intelligence*, pp. 87–117, Springer Berlin Heidelberg, 2010.

[30] R. Sarker and C. Coello Coello, "Assessment methodologies for multiobjective evolutionary algorithms," in *Evolutionary Optimization*, vol. 48 of *International Series in Operations Research & Management Science*, pp. 177–195, Springer US, 2003.

[31] M. R. Senouci and H. E. Lehtihet, "Sampling-based selection-decimation deployment approach for large-scale wireless sensor networks," *Ad Hoc Networks*, vol. 75-76, pp. 135–146, June 2018.

[32] Arduino. http://www.arduino.cc/.