IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 15, 2017, accepted May 29, 2017, date of publication June 5, 2017, date of current version June 27, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2711484

An Ant Colony Optimization Approach for the Deployment of Reliable Wireless Sensor Networks

DINA S. DEIF, (Student Member, IEEE), AND YASSER GADALLAH, (Senior Member, IEEE)
The American University in Cairo, Cairo 11835, Egypt
Corresponding author: Dina S. Deif (dinadeif@aucegypt.edu)

ABSTRACT A reliable wireless sensor network (WSN) is defined as a network that functions satisfactorily, in terms of both its coverage and connectivity to the sink(s), throughout its intended mission time. Deploying reliable WSNs is especially important for critical Internet of Things (IoT) applications, such as industrial, structural health-monitoring, and military applications. In such applications, failure of the WSN to carry out its required tasks can have serious effects, and hence, cannot be tolerated. However, the deployment of reliable WSNs is a challenging problem. This is primarily attributed to the fact that sensor nodes are subject to random failures due to different factors, such as hardware failures, battery depletion, harsh environmental conditions, and so on. In this paper, the problem of deploying a WSN with a specified minimum level of reliability at a minimum deployment cost is addressed. This problem is coined the minimum cost reliability constrained sensor node deployment problem (MCRC-SDP). The MCRC-SDP is proved to be an **NP**-Complete. An ant colony optimization algorithm coupled with a local search heuristic is proposed as a solution. Extensive experimental results demonstrate the effectiveness of the proposed approach in finding high-quality solutions to the problem.

INDEX TERMS Wireless sensor networks, sensor node deployment, network reliability, ant colony optimization, greedy heuristic, local search.

I. INTRODUCTION

Over the past decade, rapid advances in the fabrication of wireless Sensor Nodes (SNs) have broadened the range of applications of Wireless Sensor Networks (WSNs) to include residential, industrial, commercial, healthcare and military applications. As a result, WSNs have become one of the key technologies for realizing the Internet of Things (IoT) concept, playing the pivotal role of detecting events and measuring physical and environmental quantities of interest [1]. It is currently estimated that the WSN market will grow to $1.8 billion by 2024 [2]. Some of the important IoT applications place stringent reliability requirements on the WSN. For example, reliability of the WSN is considered one of the most essential attributes for industrial applications [3]. In such applications, the failure of the network to carry out its required tasks can have serious effects and hence cannot be tolerated.

A *reliable* WSN is defined as a network that functions satisfactorily, in terms of both its coverage of the targeted Region of Interest (RoI) and its connectivity to the sink(s), throughout its intended *mission time*. In other words, the WSN provides a *connected cover* of the targeted RoI throughout its mission time. The mission time for a WSN is application-dependent and can either be the intended lifetime of the network or the time interval between regular network maintenance operations. However, the deployment of reliable WSNs is a challenging problem. This is primarily attributed to the fact that SNs are subject to random failures due to different factors, such as hardware failures, battery depletion, harsh environmental conditions, etc. [4]. Hence, to guarantee the reliable operation of a WSN during its intended mission time, the presence of redundant SNs in the network becomes essential. However, for many applications for which SNs are equipped with expensive hardware, minimizing the total deployment cost remains a primary concern. Therefore, the level of SN redundancy in the WSN must be carefully quantified, such that the network meets the minimum reliability requirements imposed by the application while avoiding an unnecessary increase in the network deployment cost.

Reviewing the literature on the topic of WSN deployment reveals that there is a considerable amount of research on the problem of minimizing the deployment cost of WSNs under different assumptions and for different types of applications [5]. This problem is formulated as a constrained combinatorial optimization problem, where the objective function is the network deployment cost and the constraints are that the deployed network meets the application's coverage and connectivity requirements under the given assumptions. This optimization problem has been proven to be NP-Complete even under ideal assumptions [6]. A large number of studies [7]–[14] have proposed different algorithms to solve this problem, i.e. to find a *connected cover* of the targeted RoI that is *cost-optimal*. However, the functionality of a WSN deployment which consists of a single cost-optimal connected cover cannot be guaranteed throughout a given network mission time. This is because, by definition, a cost-optimal connected cover has no or little SN redundancy, and hence the occurrence of SN failures during the network mission time will compromise the functionality of the network in terms of coverage and/or connectivity.

Recent studies have addressed WSN fault-tolerance in conjunction with SN deployment e.g. [15]–[18]. In [15], the authors address the problem of Grid-based Coverage with Low-cost and Connectivity-guarantee (GCLC). They propose an Ant Colony Optimization (ACO) algorithm which attempts to find cost-optimal connected covers of a grid-based RoI. The proposed ACO also attempts to alleviate the energy-hole problem, which is the problem of fast energy depletion of SNs near the sink node. For this, the proposed ACO may add redundant SNs (near the sink node) to a cost-optimal connected cover based on a simple SN energy load metric. The study in [15] is extended in [16], where the proposed ACO is coupled with a deterministic algorithm. The proposed deterministic algorithm adds one or more redundant SNs to a cost-optimized connected cover to avoid the energy-hole problem around any heavily loaded SN. However, besides assuming grid-based RoIs only, both studies do not consider the fact that SN failures may occur due to factors other than energy depletion such as random hardware failures. That is, the failure of any of the SNs which are not heavily loaded in the proposed deployment strategies will still compromise the functionality of the WSN in terms of coverage and/or connectivity. The study in [17] focuses on finding cost-optimal k-coverage regular patterns for SN deployment for $4 \leq k \leq 9$. The authors propose a framework to generate cost-optimal regular deployment patterns from regular tessellations of a 2-D plane. There are several limitations to this study. First, connectivity is not addressed as part of the WSN functionality. Second, the results are only applicable to area coverage and not to target coverage which is the type of coverage required in many WSN applications (i.e. industrial monitoring). Finally, the study assumes that SNs can be deployed in *regular* patterns, which in turn means that an SN can be deployed on any point within the RoI. This assumption is invalid for many WSN applications where some locations/points in the RoI are not accessible or feasible for SN deployment. The study in [18] proposes a Genetic Algorithm (GA) which attempts to find cost-optimal deployments that satisfy $k-$coverage and $m-$connectivity. The authors assume that target coverage is required and that there are a finite number of locations in the RoI where an SN can be deployed. However, the study does not explicitly consider the reliability of the network as a deployment objective and hence does not link the deployment cost with the expected failure rates of the SNs used in deployment.

Therefore in order to deploy cost-efficient reliable WSNs, it is important to use a deployment strategy that considers network reliability *explicitly* as a design requirement while ensuring that the deployment cost is minimized. Considering the reliability requirement offers a method to predict the level of SN redundancy required to *maintain* the WSN functionality (in terms of both its coverage of the RoI and its connectivity to the sink node) *throughout* its mission time based on the failure rates of its constituent SNs. To the best of our knowledge, considering WSN reliability explicitly as a design requirement for deployment in conjunction with cost-minimization has not been addressed before in the WSN literature.

Devising this deployment strategy is the main objective of this paper. To achieve this objective, the ability to measure the reliability of a given WSN is required, i.e. a reliability *metric* is required. Recent studies have proposed different reliability metrics for WSNs subject to random SN failures [19]–[21]. In [19], the authors present a reliability metric for message delivery in WSNs. The proposed metric is derived for WSNs of a specific clustered configuration, where each cluster is assumed to monitor a specific target location in the RoI. They define the reliability of message delivery between a sink node and a given cluster as the probability that there exists a functional multi-hop wireless path between the sink node and at least one operational SN in that cluster. In [20], the authors propose a reliability metric for WSNs using a combinatorial approach [22]. They adopt the practical assumption that the WSNs can have an arbitrary configuration where SNs can monitor multiple target locations in the RoI and that each target location can be monitored by multiple SNs. However, they define the network functionality in terms of the degree of target coverage only. Connectivity of SNs with a designated sink node is not considered. The study in [21] extended the reliability metric proposed in [20] by redefining network functionality to include the connectivity between SNs and the sink node.

In this paper, we focus on the problem of deploying a WSN that meets a specified minimum level of reliability (as required by the application at hand) defined over a given mission time in such a way that results in the minimum network deployment cost. We coin this problem the *Minimum Cost Reliability-Constrained Sensor Node Deployment Problem* (MCRC-SDP). We mathematically formulate the MCRC-SDP as a combinatorial optimization problem and prove that it is **NP**-Complete. Based on the promising performance of ACO coupled with Local Search (LS) heuristics in solving complex combinatorial optimization problems [23]–[25], we propose an ACO-based approach to solve our deployment problem. ACO is a metaheuristic optimization approach in which a group or "colony" of artificial agents or "ants" cooperate using the concept of *stigmergy* (i.e. the indirect communication between the artificial ants mediated by their environment in the form of a chemical called *pheromone*) to find high quality solutions to intractable combinatorial optimization problems. To measure the reliability of the network, we adopt the reliability metric proposed in [20] with the extended functionality definition in [21], which includes both the coverage and connectivity aspects of the network and is suitable for any arbitrary configuration of the network. To the best of our knowledge, the proposed algorithm is the first algorithm to solve the MCRC-SDP.

The rest of this paper is organized as follows. In Section II, the MCRC-SDP is formally defined. The proposed ACO-based approach for solving the stated problem is proposed in Section III. In Section IV, the experimental results are presented and discussed. Finally, the paper is concluded in Section V.

II. PROBLEM DEFINITION

In this section, the MCRC-SDP is formally defined as a combinatorial optimization problem. We start by defining the WSN model in Section II.A. We then present the adopted WSN reliability metric in Section II.B. Finally, we mathematically formulate the MCRC-SDP and prove that it is **NP**-Complete in Sections II.C – II.D, respectively.

A. WIRELESS SENSOR NETWORK MODEL

We assume that the RoI is modeled as a two-dimensional area in which there is a finite set of locations that require some form of monitoring (e.g. motion, image...etc.) using static SNs. These locations are called *target points* and they represent the vital locations or assets that require monitoring in the RoI. We denote the set of target points $\boldsymbol{T} = \{t_1, t_2, \ldots, t_{|\boldsymbol{T}|}\}$. We assume that there is a finite set of possible deployment locations for SNs, which we call *deployment points*, at which SNs may be deployed. This assumption is valid for most critical WSN applications, where the topology or layout of the targeted RoI is known prior to the WSN deployment. Hence, careful examination of that RoI yields a finite set of feasible possible deployment locations, i.e. deployment points. We denote the set of deployment points $\boldsymbol{D} = \{d_1, d_2, \ldots, d_{|\boldsymbol{D}|}\}$.

All SNs available for deployment are assumed to be able to communicate wirelessly and have the same fixed communication range denoted by r_c. Sensed data acquired by the deployed SNs are relayed to a sink node with an arbitrary fixed position in the RoI denoted by d_0.

B. WIRELESS SENSOR NETWORK RELIABILITY METRIC

As discussed in Section I, all types of SNs are prone to random failures during the network mission time, denoted by $\mathcal{T}$, due to a variety of factors. Accordingly, each SN can be modeled as a two-state device, where the states are *on* and *off*. An SN in the *on* state is functional in terms of both sensing its surrounding environment (i.e. coverage) and communicating wirelessly with its neighbors (i.e. connectivity). On the other hand, an SN in the *off* state is assumed to have failed permanently in terms of both its coverage and connectivity.

Let the set of deployed SNs in a WSN be denoted by $\boldsymbol{S}=\{d_k\}, k \in [1, 2, \ldots, |\boldsymbol{D}|]$. The set $\boldsymbol{S}=\{d_k\}$ is a connected cover of a given set of target points $\boldsymbol{T}$, where $\boldsymbol{S} \subseteq \boldsymbol{D}$ and $d_k \in \boldsymbol{S}$ are the deployments points where SNs are actually deployed. Similar to [19]–[21], we assume that each deployed SN, denoted by its deployment location d_k, is characterized by a failure probability denoted by λ_k. This failure probability is equal to the probability that a given SN d_k will fail during $\mathcal{T}$, i.e. the probability that the state of d_k will change from *on* to *off* during $\mathcal{T}$. This probability can be estimated for a given $\mathcal{T}$ using the reliability function of a given SN type, which is in turn modeled using rigorous reliability testing techniques and/or gathering empirical data on the used SN type [26], [27].

Following the combinatorial approach [22] in evaluating reliability adopted in [20] and [21], the reliability of a given WSN $\boldsymbol{S}$, which is the probability that the WSN remains functional during $\mathcal{T}$, by can be expressed as follows:

$$R(\boldsymbol{S}) = \sum\nolimits_{\boldsymbol{X} \subseteq \boldsymbol{S}} [f(\boldsymbol{X}) * Prob(\boldsymbol{X})], \tag{1}$$

where $\boldsymbol{X}$ is a given *state* of the network and $f(\boldsymbol{X})$ is the network *structure function* value at that given state. The network state $\boldsymbol{X}$ is defined as the subset of SNs in $\boldsymbol{S}$ that fail during $\mathcal{T}$. Assuming that SNs fail independently, the probability of a given network state $\boldsymbol{X}$ is given by:

$$Prob\,(\boldsymbol{X}) = \prod_{d_k \in \boldsymbol{X}} \lambda_k * \prod_{d_k \in (\boldsymbol{S}-\boldsymbol{X})} (1 - \lambda_k) \tag{2}$$

On the other hand, the network structure function $f(\boldsymbol{X})$ is defined as the binary function that denotes whether the WSN is functional at a given network state $\boldsymbol{X}$ ($f\,(\boldsymbol{X}) = 1$), or not ($f\,(\boldsymbol{X}) = 0$). In this paper, we adopt the network functionality definition in [21], which assumes that the WSN is functional at a given state $\boldsymbol{X}$ if the following two conditions are met:

1. Each target point $t_\ell \in \boldsymbol{T}$, $\ell = 1, \ldots, |\boldsymbol{T}|$ is within the coverage range of at least one functional SN. Let $\boldsymbol{Y}_\ell(\boldsymbol{X})$ be the set of functional SNs covering the target point t_ℓ. Accordingly, this condition can be expressed as, $|\boldsymbol{Y}_\ell(\boldsymbol{X})| \neq 0, \forall \ell = 1, \ldots, |\boldsymbol{T}|$ where $\boldsymbol{Y}_\ell(\boldsymbol{X}) \subseteq (\boldsymbol{S} - \boldsymbol{X})$.
2. In each set $\boldsymbol{Y}_\ell(\boldsymbol{X})$, there is at least one SN $d_k \in \boldsymbol{Y}_\ell(\boldsymbol{X})$ that has at least one single- or multi-hop path to the sink node, d_{sink}. This implies that the SNs along this path are functional. Hence, the events detected at any target point t_ℓ can be relayed back to the sink node. Let the set $\boldsymbol{Z}_\ell(\boldsymbol{X})$ be the set of SNs $\{d_k\} \in \boldsymbol{Y}_\ell(\boldsymbol{X})$ that are connected to the sink node. Hence $\boldsymbol{Z}_\ell(\boldsymbol{X}) \subseteq \boldsymbol{Y}_\ell(\boldsymbol{X})$. This condition can be expressed as $|\boldsymbol{Z}_\ell(\boldsymbol{X})| \neq 0$, $\forall \ell = 1, \ldots, |\boldsymbol{T}|$.

In [20], the authors present an efficient depth-first-search-based algorithm to calculate the above WSN reliability metric. The proposed algorithm searches for the network states that correspond to $f(\boldsymbol{X}) = 1$, i.e. the tolerable SN failure combinations. The proposed algorithm significantly reduces the required number of network structure function evaluations needed to calculate the reliability of a given WSN deployment. For further details on the reliability metric and proposed algorithm, we refer the reader to [20]. In this paper, we use the reliability metric proposed in [20] coupled with the above network functionality definition to evaluate the reliability of a WSN deployment.

C. THE MINIMUM COST RELIABLITY CONSTRAINED SENSOR NODE DEPLOYMENT PROBLEM

As discussed earlier in Section I, this paper addresses the problem of deploying a WSN that meets a specified minimum level of reliability, denoted by R_{min}, defined over a given mission time at the minimum network deployment cost. The reliability requirement of the MCRC-SDP implicitly includes *three* sub-requirements. The first two sub-requirements are the fulfillment of the coverage and connectivity functionality aspects according to the WSN model presented in section II.A and the network functionality definition presented in section II.B. The third sub-requirement is that the WSN must possess a certain level of robustness against the random failures of its constituent SNs. This robustness, in turn, requires introducing a certain level of SN redundancy in the network deployment. However, SN redundancy in a given network can greatly increase the energy consumption of the network, the demand on the limited bandwidth and the level of internal interference [28], assuming that all the deployed SNs are activated at the same time, thus defeating the purpose of introducing the redundancy in the first place. Therefore, some form of SN activity planning is required to increase the fault tolerance of the WSN without introducing any degradation in its performance.

As such, we can restate the MCRC-SDP to be the problem of finding a number of non-overlapping *minimal* connected covers of the targeted RoI such that the combined reliability level of these minimal connected covers would meet or exceed the specified minimum level of reliability R_{min} and the total number of deployed SNs (i.e. the deployment cost) is minimized. A *minimal* connected cover is defined as a connected cover which contains no redundant SNs. These minimal connected covers are activated in an orthogonal manner as follows: a single minimal connected cover is activated at any given point in time during $\mathcal{T}$ while the SNs belonging to the remaining connected covers are put in sleep mode. Since there are no redundant SNs in a minimal connected cover, energy consumption, bandwidth usage and internal interference are kept at a minimum. This activated minimal connected cover remains active until its functionality is compromised due to the expected random failures of its constituent SNs. At that point, the remaining functional SNs belonging to this minimal connected cover are put in sleep mode and another minimal connected cover is activated. This procedure is continued until either the mission time of the network $\mathcal{T}$ elapses or there are no remaining deployed minimal connected covers of uncompromised functionality. The first event means the WSN deployment remained functional throughout $\mathcal{T}$ while the second event means that the WSN has failed. According to the statement of the problem, the probability of the first event is equal to R_{min} and that of the second event is equal to $1 - R_{min}$.

Let $\mathcal{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N\}$ be the superset of N non-overlapping connected covers in a given WSN deployment. For simplicity, we assume here that the WSN is homogeneous, i.e. composed of the same type of SNs. The MCRC-SDP can then be formulated as follows:

$$min \left\{ |\mathcal{S}| = \sum_{k=1}^{N} |\boldsymbol{S}_k| \right\} \tag{3}$$

Subject to:

$$\boldsymbol{S}_k \subseteq \boldsymbol{D} \quad \forall k = 1, \ldots N, N \leq N_{UB} \tag{4}$$

$$\boldsymbol{S}_k \bigcap \boldsymbol{S}_{k'} = \varphi, \forall k, k' = 1, \ldots, N, k \neq k', \tag{5}$$

$$R(\mathcal{S}) = R(\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N) = 1 - \prod_{k=1}^{N} (1 - R(\boldsymbol{S}_k)) \geq R_{min}, \tag{6}$$

$$\Phi(\boldsymbol{S}_k) = 0 \quad \forall k = 1, \ldots, N, \tag{7}$$

The objective function of the MCRC-SDP expressed in (3) is simply the minimization of the total number of deployed SNs belonging to all N non-overlapping minimal connected covers, i.e. $|\mathcal{S}|$, which corresponds to minimizing the total deployment cost of the network. Equations (4)-(7) express the constraints of the optimization problem. Equation (4) constrains all the connected covers to be subsets of $\boldsymbol{D}$, earlier defined as the set of possible SN deployment locations in the targeted RoI. Equation (4) also sets the number of connected covers N to be less than or equal to N_{UB}, which is defined as the the upper bound on the number of connected covers for a given MCRC-SDP instance, i.e. for a given $\{\boldsymbol{T}, \boldsymbol{D}\}$ tuple. The process of estimating this upper bound is detailed in Section II.D. Equation (5) expresses the disjoint property imposed on the N connected covers, while equation (6) expresses the reliability constraint of the problem. In (6), the total reliability of the WSN deployment, i.e. of $\mathcal{S}$, is calculated in terms of the reliability of the N connected covers assuming they are activated orthogonally. Finally, (7) further constrains each of the N connected covers in $\mathcal{S}$ to be a minimal connected cover, i.e. to contain no redundant SNs, where $\Phi(\mathcal{S}_k)$ is a binary function that returns 0 if the connected cover $\mathcal{S}_k$ is a minimal connected cover and 1 otherwise.

Note that if $\boldsymbol{S_k}$ is a *minimal* connected cover, the only state of $\boldsymbol{S_k}$ that would correspond to a unity network structure function (i.e. $f(\boldsymbol{X}) = 1$) is the state of no SN failures (i.e. $\boldsymbol{X} = \varphi$). This is because a minimal connected cover has no redundant SNs and hence the failure of one or more SNs would compromise the coverage and/or the connectivity conditions defined in Section II.B.

D. UPPER BOUND OF THE NUMBER OF CONNECTED COVERS N

For a given MCRC-SDP instance, i.e. for a given $\{\boldsymbol{T}, \boldsymbol{D}\}$ tuple, the upper bound for the number of connected covers (minimal or non-minimal) cannot exceed the upper bound for the number of covers, i.e. SN sets which meet the coverage constraint only. Therefore, we can use the upper bound of covers as the upper bound of the number of connected covers, which is denoted by N_{UB}. Although finding the maximum number of covers for a given MCRC-SDP instance is an **NP**-complete problem [25], we can estimate the upper bound of the number covers with the following method. Assume that all deployment points in $\boldsymbol{D}$ have SNs deployed on them. Next, find the target point in $\boldsymbol{T}$ that has the least number of SNs that cover it. We will call this target point a *critical target point*. The upper bound for the number of covers, and hence the upper bound on the connected covers N_{UB} as well, is therefore equal to the number of SNs covering the critical target point. This is because a cover of the RoI cannot provide full coverage of $\boldsymbol{T}$ without providing coverage of the critical target point. Hence, the maximum number of covers cannot exceed the number of SNs covering the critical target point when each cover contains a single SN covering the critical target point. To illustrate this, Fig. 1 shows an RoI with $\boldsymbol{T} = \{t_1, t_2, t_3\}$ and $\boldsymbol{D} = \{d_1, d_2, d_3, d_4, d_5, d_6\}$. As can be observed from the figure, target points t_1 and t_3 are both covered by 3 SNs located on deployment points d_1, d_2, d_6 and d_2, d_3, d_5 respectively, while target point t_2 is covered by only two SNs deployed on deployment points $\{d_3$ and $d_4\}$. Therefore, t_2 is the critical target point and the upper bound of the number of connected covers N_{UB} for this example is 2.

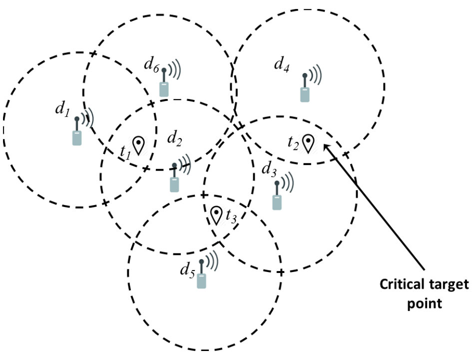


FIGURE 1. **A Region of Interest containing three target points $T = \{t_1, t_2, t_3\}$ and six possible SN deployment points $D = \{d_1, d_2, d_3, d_4, d_5, d_6\}$. Assuming SNs are deployed on all six deployment points, we find that upper bound for connected covers is $N_{max} = 2$.**

E. CLASSIFICATION OF MCRC-SDP AS AN OPTIMIZATION PROBLEM

The MCRC-SDP expressed in (3)–(7) is a combinatorial constrained optimization problem. In this section, we prove that the MCRC-SDP is **NP**-complete. To prove that, we start by considering the decision problem that corresponds to the MCRC-SDP. We will call this the decision problem the Reliability Constrained SN Deployment Problem (RC-SDP). The RC-SDP can be expressed as follows:

RC-SDP: given $\boldsymbol{D}$, $\boldsymbol{T}, N_{UB}$ ($N_{UB} \in \boldsymbol{Z}^{+}$), R_{min} ($R_{min} \in [0, 1]$), Does a superset $\mathcal{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N\} \subseteq D$ exists, such that the following conditions are true?

1. $\boldsymbol{S}_k \subseteq \boldsymbol{D} \quad \forall k = 1, \ldots N, N \leq N_{UB}$
2. $\boldsymbol{S}_k \bigcap \boldsymbol{S}_{k'} = \varphi, \quad \forall k, k' = 1, \ldots, N, k \neq k'$;
3. $R(\mathcal{S}) = 1 - \prod_{k=1}^{N} (1 - R(\boldsymbol{S}_k)) \geq R_{min}$;
4. $\Phi(\boldsymbol{S}_k) = 0 \quad \forall k = 1, \ldots, N$.

Theorem 1: RC-SDP is **NP**

Proof: $\because$ It is straightforward to prove that for any given superset $\mathcal{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N \subseteq \boldsymbol{D}$ the problem's conditions can be checked in polynomial time to decide if the corresponding answer/output to any given superset $\mathcal{S}$ is a *YES*. The computational complexity of checking each of the problem's conditions is given, in order, as follows:

1. $O(|\boldsymbol{D}|)$, where $\sum_{k=1}^{N} |\boldsymbol{S}_k| = |\mathcal{S}| \leq |\boldsymbol{D}|$;
2. $O\left(max_k \left\{|\boldsymbol{S}_k|^2\right\}\right), k = 1, \ldots, N$;
3. $O(max_k \left\{|\boldsymbol{S}_k|^3\right\})$;
4. $O(max_k \left\{|\boldsymbol{S}_k|^3\right\})$

The computational complexity of checking the third and fourth conditions is the same since checking whether a connected cover $\boldsymbol{S}_k$ is minimal or not comes automatically through calculating its reliability $R(\boldsymbol{S}_k)$. The computational complexity of calculating $R(\boldsymbol{S}_k)$ is dictated by the complexity of the structure function evaluation, which is the most computationally expensive routine in the search algorithm presented in [20] and [21]. Checking the two network functionality conditions, i.e. checking the network coverage of the set of target points $\boldsymbol{T}$ and the connectivity to the sink at a given network state, has a computational complexity of $O(|\boldsymbol{S}_k| * |\boldsymbol{T}|)$ and $O(|\boldsymbol{S}_k|^3)$, respectively. This gives an overall computational complexity of $O(max_k \left\{|\boldsymbol{S}_k|^3\right\})$, where $0 < |\boldsymbol{S}_k| \leq |\boldsymbol{D}|$.

$\therefore$ RC-SDP is **NP** ■

Theorem 2: RC-SDP is **NP**-hard.

Proof: Using the method of restriction, we let $N_{UB} = 1$ and $R_{min} = \epsilon \ll 1$. This means that we are looking at a single connected cover deployment and that any non-zero value of reliability is acceptable. This restriction converts the RC-SDP to the problem of deciding whether there exists a single connected cover $\boldsymbol{S} \subseteq \boldsymbol{D}$ of size/cardinality $|\boldsymbol{S}| \leq |\boldsymbol{D}|$ that provides full coverage of $\boldsymbol{T}$ and is connected to the given sink node. This latter problem has been proved **NP**-complete in [29].

$\therefore$ RC-SDP is **NP-**hard ■

From theorems 1, 2 $\rightarrow$ RC-SDP is **NP**-complete ■

III. ANT COLONY OPTIMIZATION BASED APPROACH FOR DEPLOYING RELIABLE WIRELESS SENSOR NETWORKS

In this section, we present our proposed ACO approach for solving the MCRC-SDP expressed in (3)–(7). First, we discuss how the MCRC-SDP is represented as a connected graph for ACO application, i.e. define the *construction graph* of the problem. Then, the ants' tour construction procedure is described, including the ants' neighborhood definitions and heuristic information. This is followed by the formulation of the cost function used for evaluating the quality of the solutions obtained by the ants. We then describe the pheromone management scheme followed by the LS procedure which we propose to be coupled with the ACO algorithm to enhance the quality of the obtained solutions. Finally, we summarize the steps of the proposed algorithm.

A. CONSTRUCTION GRAPH

In any ACO algorithm designed to solve a given optimization problem, ants build solutions incrementally by executing randomized walks or *tours* through a connected graph $G(\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the set of the graph's vertices and $\boldsymbol{E}$ is the set of all the edges between the vertices in $\boldsymbol{V}$. Therefore, the first step in designing an ACO algorithm to solve a given optimization problem is to represent the problem as a connected graph $G(\boldsymbol{V}, \boldsymbol{E})$ by defining the sets $\boldsymbol{V}$ and $\boldsymbol{E}$ in terms of the problem's variables. For the MCRC-SDP at hand, the ACO construction graph is identical to the problem's graph defined by the set of deployment points $\boldsymbol{D}$ and the location of the sink node denoted by d_0. Hence, $\boldsymbol{V}$ corresponds to the set of possible deployment points and the sink node location (i.e. $\boldsymbol{V} \equiv \{d_0, \boldsymbol{D}\} = \{d_0, d_1, d_2, \ldots, d_{|\boldsymbol{D}|}\}$) and $\boldsymbol{E}$ corresponds to the set of undirected arcs/links connecting the deployment points and the sink node in $\boldsymbol{V}$ with each other.

B. TOUR CONSTRUCTION

The ants' search behavior in a given construction graph is primarily influenced by a probabilistic transition rule, which controls how each ant selects its next vertex (i.e. deployment point) to visit during the construction of its tour (i.e. its solution to the problem). The probabilistic transition rule is in turn defined by three elements: the neighborhood definition(s), the heuristic information used by the ant and the pheromone trail values between the vertices of the construction graph. In this section we will discuss the first two elements while the pheromone management is discussed in Section III.E.

1) BASIC IDEA

Each ant a, $a = 1, \ldots, m$, starts its tour at the sink node location d_0, which is an arbitrary location inside the boundaries of the RoI. Let the solution to the problem at hand which corresponds to the ant's tour be denoted by $\mathcal{S}^a$, initialized by an empty superset, i.e. $\mathcal{S}^a = \varphi$. Ant a then starts constructing a solution to the problem by consecutively building connected covers through transitioning among the deployment points in the construction graph. Let the index of the connected covers built by ant a be denoted by k, where $k = 1$ in the beginning of the ant's tour. An SN is deployed at each deployment point visited by ant a and the deployment point is added to the connected cover that ant a is currently building, denoted by $\boldsymbol{S}_k^a$. The connectivity of $\boldsymbol{S}_k^a$ to the sink node is maintained in each ant's transition by selectively defining the neighborhood of the ant's probabilistic transition rule (i.e. the candidate deployment points selected for the next transition), which will be discussed in the next sub-section. The building of $\boldsymbol{S}_k^a$ concludes when complete coverage of the target points in set $\boldsymbol{T}$ is achieved. The completed connected cover $\boldsymbol{S}_k^a$ is then added to the ant's solution superset $\mathcal{S}^a$. To check if the ant's tour is complete, $R(\mathcal{S}^a)$ is calculated using (6) and compared to the given minimum reliability level R_{min}. If $R\left(\mathcal{S}^a\right) \geq R_{min}$, then ant a's tour is concluded. Otherwise, the index k is incremented and ant a starts building a new connected cover through transitioning between the deployments points in the construction graph, excluding the points belonging to the connected cover(s) the ant built and added to $\mathcal{S}^a$ so far. Ant a continues building connected covers until $R\left(\mathcal{S}^a\right)$ meets or exceeds R_{min}. At this point the solution corresponding to ant a's tour is denoted $\mathcal{S}^a = \{\boldsymbol{S}_1^a, \boldsymbol{S}_2^a, \ldots, \boldsymbol{S}_{N_a}^a\}$.

2) HEURISTIC INFORMATION AND NEIHBOURHOOD DEFINITIONS

At each tour construction step, ant a applies a probabilistic transition rule to select which deployment point it will visit next. The probability that ant a, currently at deployment point $d_i, i = 0, 1, \ldots, |\boldsymbol{D}|$, will select deployment point $d_j, j = 1, 2, \ldots |\boldsymbol{D}|$, to visit next is given by:

$$p_{ij}^a = \begin{cases} \dfrac{[\tau_{ij}]^\alpha [\eta_j^a]^\beta}{\sum_{d_\ell \in \mathcal{N}_i^a} [\tau_{il}]^\alpha [\eta_\ell^a]^\beta}, & \text{if } d_j \in \mathcal{N}_i^a \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where τ_{ij} is the pheromone trail value between deployment points d_i (or sink node if $i = 0$ at the beginning of the tour) and d_j, η_j^a is the heuristic value of adding the deployment point d_j to the connected cover currently being built by ant a, i.e. $\boldsymbol{S}_k^a$, $\mathcal{N}_i^a$ is the feasible neighborhood of ant a at its current position in the construction graph at d_i, and α and β are parameters that control the influence of the pheromone trail values and heuristic information on p_{ij}^a, respectively.

The definition of the feasible neighborhood $\mathcal{N}_i^a$ of ant a at a given current position d_i depends on whether the next transition the ant is making is an *intra-connected cover transition* or *inter-connected cover transition*. Ant a makes an intra-connected cover transition when the current connected cover its building, i.e. $\boldsymbol{S}_k^a$, is not yet complete after the addition of the deployment point d_i at which the ant is currently present, i.e. $\boldsymbol{S}_k^a \neq \varphi$. On the other hand, ant a makes an inter-connected cover transition when its previous transition has completed $\boldsymbol{S}_k^a$ but its tour is not yet complete, i.e. $R\left(\mathcal{S}^a\right) < R_{min}$. In this case, the next transition of a is the start of a new connected cover, i.e. $k = k + 1$ and $\boldsymbol{S}_k^a = \varphi$.

For an intra-connected cover transition, the feasible neighborhood $\mathcal{N}_i^a$ of ant a at a given current position d_i is defined as follows:

$$\mathcal{N}_i^a = \begin{cases} \mathcal{N}_{ieff}^a, & \mathcal{N}_{ieff}^a \neq \varphi \\ \mathcal{N}_{ifull}^a, & \mathcal{N}_{ieff}^a = \varphi, \end{cases} \tag{9}$$

where $\mathcal{N}_{ifull}^a$ is defined as the set of deployment points within the communication range r_c of any deployment point belonging to $\boldsymbol{S}_k^a$. Let the set $\boldsymbol{D}^-$ be the set of deployment points not visited so far by ant a in its current tour. The set $\mathcal{N}^a_{ifull}$ can then be expressed as follows:

$$\mathcal{N}^a_{ifull} = \{d_j \in \boldsymbol{D}^- : \left\| d_j d_{j'} \right\| \leq r_c, \text{ for any } d_{j'} \in \boldsymbol{S}^a_k\} \quad (10)$$

The set $\mathcal{N}^a_{ieff}$, on the other hand, is a subset of deployment points belonging to $\mathcal{N}^a_{ifull}$ that would offer a *coverage gain* for $\boldsymbol{S}^a_k$, i.e. the addition of any of the deployment points belonging to $\mathcal{N}^a_{ieff}$ to $\boldsymbol{S}^a_k$ would result in the coverage of uncovered target points in $\boldsymbol{T}$ by $\boldsymbol{S}^a_k$. Let the coverage gain of a deployment point $d_j \in \mathcal{N}^a_{ifull}$ be denoted by g^a_j. We define the coverage gain g^a_j as the number of uncovered target points by $\boldsymbol{S}^a_k$ that would be covered if an SN is deployed at d_j, i.e. if d_j is added to the current connected cover $\boldsymbol{S}^a_k$. Hence, the set $\mathcal{N}^a_{ieff}$ can be expressed as follows:

$$\mathcal{N}^a_{ieff} = \{d_j \in \mathcal{N}^a_{ifull} : g^a_j \neq 0\} \quad (11)$$

For an inter-connected cover transition, on the other hand, the feasible neighborhood $\mathcal{N}^a_i$ of ant a at a given current position d_i is defined as:

$$\mathcal{N}^a_i = \mathcal{N}^a_{sink}, \quad (12)$$

where $\mathcal{N}^a_{sink}$ is defined as the set of deployment points belonging to $\boldsymbol{D}^-$ which are within a distance equal to the SN communication range r_c. Note that at the beginning of the tour, $i = 0$ and $\boldsymbol{D}^- = \boldsymbol{D}$. Accordingly, we can express $\mathcal{N}^a_{sink}$ as follows:

$$\mathcal{N}^a_{sink} = \{d_j \in \boldsymbol{D}^- : \left\| d_j d_0 \right\| \leq r_c\} \quad (13)$$

The neighborhood definitions in (9) and (12) are designed to achieve two goals. The first goal is to guarantee the connectivity of each cover built by ant a. Since all ants start their tours at d_0, the neighborhood definitions guarantee that each added deployment point to $\boldsymbol{S}^a_k$ will be connected to the sink node via single or multi-hop communication. The second goal is to minimize the probability of adding redundant deployment points to any of the connected covers built by the ants, i.e. minimize the probability of ants constructing tours that correspond to infeasible solutions to the MCRC-SDP that violate the redundancy constraint expressed in (7). This goal is achieved specifically through the neighborhood definition in (9). The neighborhood definition restricts the candidate deployment points for the ant's next transition to points which belong to the set $\mathcal{N}^a_{ifull}$ and have a non-zero coverage gain, i.e. $\mathcal{N}^a_{ieff}$. In the case where $\mathcal{N}^a_{ieff} = \varphi$, however, adding a redundant deployment point to $\boldsymbol{S}^a_k$ may occur.

The heuristic value of adding deployment point d_j to a current connected cover $\boldsymbol{S}^a_k$ being built by ant a, denoted by η^a_j, is directly proportional to its coverage gain g^a_j and is defined as:

$$\eta^a_j = g^a_j + 1 \quad (14)$$

Equation (14) applies to both types of ant's transitions, namely, the intra- and inter-connected cover transitions, where in the latter case the uncovered target points are the entire set $\boldsymbol{T}$, since the current connected cover $\boldsymbol{S}^a_k$ in this case is empty, i.e. $\boldsymbol{S}^a_k = \varphi$.

ALGORITHM 1 summarizes the ants' tour construction procedure.

Algorithm 1 Tour Construction Procedure in the Proposed ACO Algorithm

Procedure TOUR_CONSTRUCTION ($\boldsymbol{a}$)
Input: $\boldsymbol{D}, \boldsymbol{T}, d_0, R_{min}, \lambda, \tau_{ij}$ for $i = 0, 1, \ldots, |\boldsymbol{D}|, j = 1, 2, \ldots, |\boldsymbol{D}|$
Initialize: $\mathcal{S}^a = \varphi, R(\mathcal{S}^a) = 0, \quad k = 0, \boldsymbol{D}^- = \boldsymbol{D}$, ant starts tour at $d_0 (i = 0)$
While $R(\mathcal{S}^a) < R_{min}$
  Build a new connected cover: $k \leftarrow k + 1, \boldsymbol{S}^a_k = \varphi, \boldsymbol{T}_{cov} = \varphi$
  While $\boldsymbol{T}_{cov} \neq \boldsymbol{T}$ (i.e. $\boldsymbol{S}^a_k$ is not a complete connected cover)
    Identify $\mathcal{N}^a_i$ using (9) and (12)
    Calculate coverage gain $g_j \forall d_j \in \mathcal{N}^a_i$
    Apply transition rule in (8) to choose next deployment point
    Update $\boldsymbol{S}^a_k$
    Update $\boldsymbol{T}_{cov}$ (i.e. update coverage of $\boldsymbol{S}^a_k$)
  End While
  Update $\mathcal{S}^a : \mathcal{S}^a \leftarrow \{\mathcal{S}^a, \boldsymbol{S}^a_k\}$
  Calculate $R(\boldsymbol{S}^a_k)$ and Update $R(\mathcal{S}^a)$
  Update $\boldsymbol{D}^- : \boldsymbol{D}^- \leftarrow \boldsymbol{D}^- - \boldsymbol{S}^a_k$
End While
Output: $\mathcal{S}^a = \{\boldsymbol{S}^a_1, \boldsymbol{S}^a_2, \ldots, \boldsymbol{S}^a_{N_a}, R(\mathcal{S}^a)$

C. COST FUNCTION

To evaluate the quality of the solution to the MCRC-SDP corresponding to the tour constructed by ant a, i.e. $\mathcal{S}^a = \{\boldsymbol{S}^a_1, \boldsymbol{S}^a_2, \ldots, \boldsymbol{S}^a_{N_a}\}$, the following cost function is used:

$$C(\mathcal{S}^a) = \omega_1 \sum\nolimits_{k=1}^{N_a} |\boldsymbol{S}_k| + \omega_2 \sum\nolimits_{k=1}^{N_a} \Phi(\boldsymbol{S}_k), \quad (15)$$

where the first term of the cost function, $\sum_{k=1}^{N_a} |\boldsymbol{S}_k| = |\mathcal{S}^a|$, represents the total number of deployment points (i.e. deployed SNs) belonging to the N_a connected covers in $\mathcal{S}^a$ multiplied by a constant weight ω_1. The second term of the cost function penalizes *every* connected cover that contains redundancy i.e. that is not a minimal connected cover by a penalty equal to the constant weight ω_2.

Since the objective of the MCRC-SDP is to minimize the total deployment cost of the network, i.e. minimize $\sum_{k=1}^{N_a} |\boldsymbol{S}_k| = |\mathcal{S}^a|$, the weights ω_1 and $\omega_2$are set such that the cost assigned to the solutions follow the following criterion. All the solutions which meet both the reliability and the redundancy constraints expressed in (6) and (7), respectively, have a lower cost than all the solutions that meet the reliability constraint but fail to meet the redundancy constraint, i.e. solutions that have one or more non-minimal connected covers. As such, if ω_1 is set to unity such that the first term of the cost function is equal to the total number of deployed SNs (i.e. the deployment cost), then ω_2 must be greater than $|\boldsymbol{D}|$ (since the maximum value of $|\mathcal{S}^a|$ is $|\boldsymbol{D}|$). Accordingly, we set $\omega_1 = 1$ and $\omega_2 = |\boldsymbol{D}| + 1$.

D. LOCAL SEARCH PROCEDURE

As discussed earlier, the proposed ACO algorithm for solving the MCRC-SDP is coupled with an LS procedure that helps the algorithm find higher quality solutions to the problem [23], [24]. In each iteration of the algorithm, after the ants have completed the construction of their tours/solutions, the LS procedure is applied to each of the constructed solutions with the objective of reducing its cost as evaluated by the cost function in (15). ALGORITHM 2 shows the pseudo code of the proposed LS procedure.

Algorithm 2 Local Search Procedure for the Proposed Algorithm

Procedure LOCAL_SEARCH

Input: $\mathcal{S}^a = \left\{\boldsymbol{S}_1^a, \boldsymbol{S}_2^a, \ldots, \boldsymbol{S}_{N_a}^a\right\}, C(\mathcal{S}^a), R(\mathcal{S}^a)$

Initialize: $\mathcal{C}_{LS}\left(\mathcal{S}^a\right) \leftarrow C(\mathcal{S}^a), \mathcal{S}_{LS}^a \leftarrow \mathcal{S}^a, R(\mathcal{S}_{LS}^a) \leftarrow R(\mathcal{S}^a)$

$\mathcal{S}_{temp}^a \leftarrow \mathcal{S}_{LS}^a, R(\mathcal{S}_{temp}^a) \leftarrow R(\mathcal{S}_{LS}^a)$

For $k = 1, \ldots, N_a$

  If $\Phi(\boldsymbol{S}_k^a) = 1$, i.e. if $\boldsymbol{S}_k^a$ is *not* a minimal connected cover

    Prune $\boldsymbol{S}_k^a$ until there are no redundant deployment points. Let pruned $\boldsymbol{S}_k^a$ be denoted $\boldsymbol{S}_{kp}^a$

    Update $\mathcal{S}_{temp}^a : \boldsymbol{S}_k^a \leftarrow \boldsymbol{S}_{kp}^a$

    Update $R(\mathcal{S}_{temp}^a)$

    If $R(\mathcal{S}_{temp}^a) \geq R_{min}$

      Update $\mathcal{S}_{LS}^a : \boldsymbol{S}_k^a \leftarrow \boldsymbol{S}_{kp}^a$. $R(\mathcal{S}_{LS}^a) \leftarrow R(\mathcal{S}_{temp}^a)$

      $\mathcal{C}_{LS}\left(\mathcal{S}^a\right) \leftarrow \mathcal{C}_{LS}\left(\mathcal{S}^a\right) - \omega_2$

  Else

      $\mathcal{S}_{LS}^a$ remains unchanged $\rightarrow \mathcal{C}_{LS}\left(\mathcal{S}^a\right)$ remains unchanged

      $\mathcal{S}_{temp}^a \leftarrow \mathcal{S}_{LS}^a, R(\mathcal{S}_{temp}^a) \leftarrow R(\mathcal{S}_{LS}^a)$

  End If

 End If

End For

Output: $R(\mathcal{S}_{LS}^a), \mathcal{C}_{LS}\left(\mathcal{S}^a\right)$

The operation of the LS procedure can be described as follows. Assuming the LS is applied on the solution $\mathcal{S}^a = \{\boldsymbol{S}_1^a, \boldsymbol{S}_2^a, \ldots, \boldsymbol{S}_{N_a}^a\}$ constructed by ant a, the first step of the LS procedure is to determine whether any of the connected covers in $\mathcal{S}^a$ violates the redundancy constraint in (7), i.e. $\Phi(\boldsymbol{S}_k^a) = 1$, for any $k = 1, \ldots, N_a$. If all the connected covers are minimal connected covers, i.e. $\mathcal{S}^a$ is a feasible solution, the LS procedure returns $\mathcal{S}^a$ and its corresponding reliability $R\left(\mathcal{S}^a\right)$ unchanged. On the other hand, if one or more of the connected covers in $\mathcal{S}^a$ have redundant deployment points, the LS attempts to reduce the cost $C(\mathcal{S}^a)$ by converting these connected covers to minimal connected covers. This procedure is carried out as follows. For each non-minimal connected cover $\boldsymbol{S}_k^a$, the LS procedure *prunes* $\boldsymbol{S}_k^a$ by removing redundant deployment points. A redundant deployment point in $\boldsymbol{S}_k^a$ is a deployment point whose removal from the connected cover will not compromise its coverage or connectivity. Redundant deployment points can be identified by examining the tolerable failure combinations produced by the search algorithm used to calculate $R(\boldsymbol{S}_k^a)$ [20].

Let the pruned connected cover be denoted $\boldsymbol{S}_{kp}^a$. The LS procedure then updates the combined reliability of $\mathcal{S}^a$ accordingly (i.e. substituting $R(\boldsymbol{S}_k^a)$ with $R(\boldsymbol{S}_{kp}^a)$ in (6)). If the updated combined reliability of $\mathcal{S}^a$ exceeds or meets R_{min}, the pruned connected cover $\boldsymbol{S}_{kp}^a$ replaces $\boldsymbol{S}_k^a$ in the solution $\mathcal{S}^a$, otherwise $\boldsymbol{S}_k^a$ is kept without change in $\mathcal{S}^a$. The same above steps are repeated for every non-minimal connected cover in $\mathcal{S}^a$. Accordingly, for every pruned connected cover that replaces a non-minimal connected cover in $\mathcal{S}^a$, the cost $C(\mathcal{S}^a)$ is reduced by the value of $\omega_2 = |\boldsymbol{D}| + 1$.

E. PHEROMONE MANAGEMENT

After all the ants have constructed their tours and the LS procedure has been applied to the corresponding solutions, pheromone trail values are updated according to the MAX-MIN Ant System (MMAS) [30] updating rule which can be expressed as follows:

$$\tau_{ij} \leftarrow (1-\rho)\,\tau_{ij} + \Delta\tau_{ij}^{ib}, \tag{16}$$

where $i = 0, 1, \ldots, |\boldsymbol{D}|, j = 1, \ldots, |\boldsymbol{D}|$, $\rho \in (0, 1)$ is the pheromone evaporation factor and the added pheromone trail $\Delta\tau_{ij}^{ib}$ can be given by the following equation:

$$\Delta\tau_{ij}^{ib} = \begin{cases} 1/\mathcal{C}^{ib}, & \textit{if } d_j \in \mathcal{S}^{ib} \\ 0, & \textit{otherwise}, \end{cases} \tag{17}$$

where $\mathcal{S}^{ib}$ is the best solution found by the ants in the current iteration of the algorithm (i.e. iteration-best solution) and $\mathcal{C}^{ib}$ is its cost evaluated by the cost function expressed in (15). According to the MMAS pheromone update rule, only the ant which found the solution with the highest quality (i.e. the lowest cost) gets to deposit pheromone on the arcs of the construction graph.

Note that pheromone is deposited on all the arcs leading to the deployment point $d_j \in \mathcal{S}^{ib}$. This is because the proposed algorithm rewards the inclusion of a deployment point in the iteration-best solution, regardless of its position in the solution (i.e. regardless of the connected cover to which it belongs). The reasoning behind this is that the inclusion of such advantageous deployment points in different connected covers can lead to different but equally good solutions to the problem. Thus, exploring different permutations of these deployment points is essential to finding high quality solutions.

Since the MMAS pheromone update rule strongly exploits the best solution found in each iteration, upper and lower limits, denoted τ_{max} and τ_{min}, are imposed on the pheromone trail value on each arc of the construction graph. This strategy is called *pheromone constraining* and is followed to avoid a stagnation situation where the algorithm converges prematurely to good but sub-optimal solutions. This is due to the excessive increase of the pheromone trails on the arcs leading to the deployment points belonging to those solutions. Pheromone constraining ensures that the probability of an ant a on deployment point d_i selecting a deployment point $d_j \in \mathcal{N}_i^a$ is always greater than zero. The value of τ_{max} is given by:

$$\tau_{max} = 1/\rho\mathcal{C}^{bs}, \tag{18}$$

where $\mathcal{C}^{bs}$ is the best solution found so far by the algorithm (i.e. best-so-far solution). Note that every time Higher quality solution is found and $\mathcal{C}^{bs}$ is updated, the value of τ_{max} is updated accordingly. On the other hand, the value of τ_{min} is given by:

$$\tau_{min} = \tau_{max}/b, \tag{19}$$

where b is a constant that is set by experimentation.

Algorithm 3 The Proposed Ant Colony Optimization Algorithm for Solving the MCRC-SDP

```
Ant Colony Optimization Algorithm for Solving MCRC-SDP
Input: D, T, d_0, R_min, λ, N_UB, r_s, r_c, m, ρ, it_max, it_c
Initialize: it = 0, C^bs = ∞, S^bs = φ, τ_0 = 1
While (it < it_max & it_c > 0)
 Increment iterations counter: it ← it + 1
 For a = 1, ..., m
  Apply TOUR_CONSTRUCTION (a) procedure to build S^a
  Calculate tour cost C(S^a) using (15)
  Apply LOCAL_SEARCH procedure: S^a ← S^a_LS,
  C(S^a) ← C_LS(S^a), R(S^a) ← R(S^a_LS)
 End For
 Identify iteration-best solution S^ib and cost C^ib
 Update pheromone trails using (16), (17)
 If C^ib < C^bs
  Update best solution so far: S^bs ← S^ib, C^bs ← C^ib
  Re-initialize convergence counter it_c to starting value
 Else
  Decrement convergence counter: it_c ← it_c − 1
 End If
 Apply Pheromone constraining to τ_max and τ_min (18), (19)
End While
Output: S^bs = {S^bs_1, S^bs_2, ..., S^bs_Nbs}, C^bs
```

F. SUMMARY OF THE PROPOSED ALGORITHM

ALGORITHM 3 summarizes the different steps in the proposed ACO algorithm for solving the MCRC-SDP. The input to the proposed ACO algorithm includes all the MCRC-SDP instance parameters ($\boldsymbol{D}, \boldsymbol{T}, d_0, \lambda, R_{min}, N_{UB}, r_s, r_c$) and the ACO related parameters (m, ρ, it_{max}, it_c). The ACO parameters it_{max} and it_c are defined as the maximum allowed number of iterations the algorithm can carry out and the number of successive iterations the algorithm can carry out with no enhancement in the best so far solution cost $\mathcal{C}^{bs}$ before it is terminated, i.e. before it is decided that the algorithm has converged.

In the first step of the proposed algorithm, the best-so-far solution cost $\mathcal{C}^{bs}$ is initialized to a high value in order to ensure that it is replaced by the best solution cost found in the first iteration. All Pheromone trails are initialized to unity to ensure that they are constrained to the upper limit calculated at the end of the first iteration using (18). Then, each ant a, for $a = 1, \ldots, m$, constructs its tour/ solution $\mathcal{S}^a$ according to the tour construction procedure presented in Section II.B and summarized in Table 1. The cost of ant a's solution $C\left(\mathcal{S}^a\right)$ is evaluated using (15). Then the LS procedure presented in Section II.D and summarized in ALGORITHM 2 is applied to $\mathcal{S}^a$. It should be noted that if the LS procedure produced no reduction in the value of $C\left(\mathcal{S}^a\right)$, it returns the original solution and cost unaltered. After these steps are applied for each ant, the iteration-best solution $\mathcal{S}^{ib}$ and the corresponding cost $\mathcal{C}^{ib}$ are identified and used to update the pheromone trail values using (16) and (17). Next, the best-so-far solution is updated if $\mathcal{C}^{ib}$ is less than the current $\mathcal{C}^{bs}$ and the values of τ_{max} and τ_{min} are updated accordingly using (18) and (19). The pheromone constraining procedure follows as described in Section III.E. Finally, the algorithm is terminated if it goes through it_{max} iterations or if it goes through it_c iterations with no enhancement in the best-so-far solution cost $\mathcal{C}^{bs}$.

IV. EXPERIMENTAL WORK AND DISCUSSION

In this section, we conduct a series of experiments with two main objectives. The first objective is to determine the optimum setting of the parameters of the proposed ACO algorithm, specifically the settings for α and β, which control the influence of the pheromone trail values and heuristic values on the ants' probabilistic transition rule. The second objective is to evaluate the performance of the proposed ACO algorithm in solving the MCRC-SDP defined in Section II. Since, to the best of our knowledge, the proposed algorithm is the first algorithm to solve the MCRC-SDP, we benchmark its performance with a Greedy Heuristic (GH) which uses the same heuristic information and the same basic idea of solution construction that is adopted in the proposed ACO algorithm. The performance of both methods is measured in terms of two metrics: their success rate in obtaining feasible solutions for the MCRC-SDP (specifically solutions that do not violate the redundancy constraint) and the quality of the obtained solutions, i.e. the deployment cost which is equal to the total number of deployment points in the connected covers comprising the solutions.

A. EXPERIMETAL SETUP

In the conducted experiments, we generate instances of the MCRC-SDP of different problem scales and different values of the minimum required reliability R_{min}. We assume that the RoI is a two-dimensional square area equal to $100 \times 100\, m^2$. The target points in the set $\boldsymbol{T}$, the possible SN deployment points in the set $\boldsymbol{D}$ and the location of the sink node d_0 are all generated randomly inside the perimeter of the RoI. The scale of the problem is identified by the sizes of the sets $\boldsymbol{D}$ and $\boldsymbol{T}$, denoted by $|\boldsymbol{D}|$ and $|\boldsymbol{T}|$ respectively. For each problem scale, the upper bound of the number of connected covers N_{UB} is calculated using the procedure presented in Section II.D. We denote each problem scale a *test case*. We adopted a random generation of the test cases' data sets for a fair and thorough evaluation of the proposed algorithm.

Since the value of R_{min} affects the degree of difficulty of the problem (the higher the value the more difficult the problem instance), three values for R_{min} are considered, namely $R_{min} = 0.99$, 0.999 and 0.9999, for test case, i.e. each test case generates three problem instances, one for each of the three R_{min} values. Table 1 shows the data pertinent to each test case, namely, the values of $|\boldsymbol{D}|$, $|\boldsymbol{T}|$ and N_{UB}.

For all problem instances, we set $r_s = 30$ m, $r_c = 50$ m and $\lambda = 0.02$. The ACO parameters are set as follows: $\rho = 0.5$, $m = 30$, $it_{max} = 100$, $it_c = 20$ and $b = 10$.

B. TUNING THE ACO ALGORITHM PARAMETERS

The ACO parameters α and β control the effect of the pheromone trail values and heuristic values on ants' probabilistic transition rule, i.e. on the probability of next deployment point selection p_{ij}^{a} as expressed in (8). It is therefore important to find their optimum configuration that result in the best average solution quality obtained by the ACO algorithm. In ACO literature, values of both α and β can vary between 1 and 5, with the optimum configuration largely depending on the type of problem the ACO algorithm is designed to solve and whether or not the algorithm is coupled with an LS procedure [30].

TABLE 1. Test cases.

Test Case	$\lvert D \rvert$	$\lvert T \rvert$	N_{UB}
TC1	30	15	4
TC2	40	25	4
TC3	50	35	5
TC4	60	45	5
TC5	70	55	6
TC6	80	65	6
TC7	90	75	7
TC8	100	85	6

In order to find the optimum configuration of α and β for the MCRC-SDP, we selected two problem instances at random from the twenty four problem instances generated from the eight test cases in Table 1, namely test case TC3 for $R_{min} = 0.9999$ and TC6 for $R_{min} = 0.99$. For both selected problem instances, the proposed ACO algorithm is applied using twenty-five possible combinations of α and β, with each parameter ranging between 1 and 5. To account for the heuristic nature of the ACO algorithm, the algorithm is run ten independent times at each of the twenty-five parameters' settings.

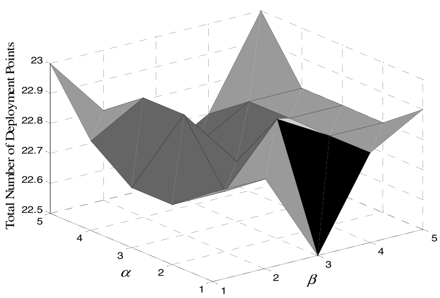


FIGURE 2. The average deployment cost obtained from applying the proposed ACO algorithm on test case TC3 at $R_{min} = 0.9999$. The best combination of α and β is $(1, 3)$.

Fig. 2 and Fig. 3 show the average value of the total number of deployment points, i.e. the average deployment cost, in the obtained solutions the versus α and β. Fig. 2 shows that there is an advantage in setting $\alpha = 1$ and $\beta = 3$, at which the minimum average number of deployment points is obtained. Fig. 3 also shows that the minimum average number of deployment points is obtained by setting $\alpha = 1$ and $\beta = 3$ in addition to setting $\alpha = 2$ and $\beta = 5$. Hence, in the following experiments we set $\alpha = 1$ and $\beta = 3$. Table 2 lists the values of the proposed ACO algorithm parameters.

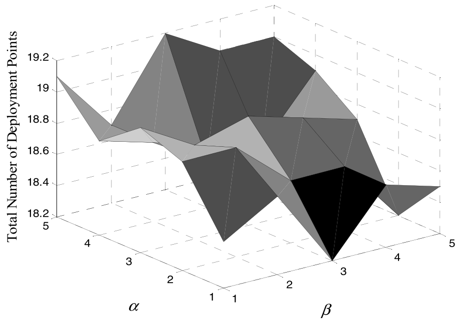


FIGURE 3. The average deployment cost obtained from applying the proposed ACO algorithm on test case TC6 at $R_{min} = 0.99$. The best combinations of α and β are $(1, 3)$ and $(2, 5)$.

TABLE 2. Parameters of the proposed ACO algorithm.

Parameter	Setting
Number of ants m	30
Pheromone influence parameter α	1
Heuristic info influence parameter β	3
Pheromone levels update rule	MMAS
Pheromone evaporation rate ρ	0.5
Pheromone constraining parameter ƀ	10
Heuristics	LS: ALGORITHM II
Maximum no. of generations it_{max}	100
No. of generations for convergence	20

C. COMPARISON WITH THE GREEDY HEURSITIC

Since, to the best of our knowledge, the proposed approach is the first algorithm for solving the minimum cost reliability constrained sensor node deployment problem, a benchmark approach is required to evaluate the performance of the proposed algorithm. Similar to the studies in [7], [13] and [25], we benchmark the performance of the proposed ACO algorithm, in terms of the quality of the obtained solutions, using a Greedy Heuristic (GH). The GH uses the same heuristic information adopted in the proposed ACO algorithm. It also follows the same basic idea of constructing solutions to the MCRC-SDP by consecutively building connected covers until the combined reliability of the connected covers meets or exceeds the specified minimum reliability R_{min}. The pseudo code of the GH is given in ALGORITHM IV.

The input to GH includes all the MCRC-SDP instance parameters: $\boldsymbol{D}$, $\boldsymbol{T}$, d_0, R_{min}, λ, N_{UB}, r_s and r_c. The GH is initialized by an empty solution superset (i.e. $\mathcal{S} = \varphi$) and a connected cover index $k = 0$. The GH then proceeds in rounds. In each round, a deployment point is added to the connected cover with the current index k, denoted by $\boldsymbol{S}_k$. Similar to the proposed ACO algorithm, connectivity to the sink node located at d_0 is achieved by constricting the candidate deployment points for inclusion to $\boldsymbol{S}_k$ in a given round to the set $\mathcal{N}_{next}$. Similar to the neighborhood set $\mathcal{N}_i^a$ defined in Section III.B and expressed in (9) and (12), the definition of the set $\mathcal{N}_{next}$ depends on whether the current round is an intra- or inter- connected cover round. For an intra-connected cover round, $\mathcal{N}_{next}$ has a similar definition to the set $\mathcal{N}^{a}_{ifull}$ expressed in (10), which includes all the deployment points which have not been added in previous rounds to the solution and are within the communication range r_c of any of the deployment points belonging to S_k. For an inter-connected cover round (which includes the first round, i.e. the start of the first connected cover), on the other hand, $\mathcal{N}_{next}$ has a similar definition to the set $\mathcal{N}^{a}_{sink}$, which includes all the deployment points which have not been added in previous rounds to the solution and are within the communication range r_c of the sink node located at d_0.

Algorithm 4 Greedy Heuristic in Comparison With the Proposed ACO Algorithm

```
Procedure GREEDY_HEURISTIC
Input: D, T, d_0, λ, R_min, N_UB, r_s, r_c
Initialize: S = φ, R(S) = 0, k = 0, D^- = D
While R(S) < R_min
 Build a new connected cover: k ← k + 1, S_k = φ, T_cov = φ
 While T_cov ≠ T (i.e. S_k is not a complete connected cover)
  If S_k = φ, i.e. the beginning of S_k
    N_next = {d_j ∈ D^- : ||d_j d_0|| ≤ r_c
  Else
    N_next = {d_j ∈ D^- : ||d_j d_j'|| ≤ r_c for any d_j' ∈ S_k
  End If
  Calculate coverage gain g_j ∀d_j ∈ N_next
  UpdateS_k by adding d_j ∈ N_next with g_j = g_max
  Update T_cov (i.e. update coverage of S_k)
 End While
 Update S : S ← S ∪ S_k
 Calculate R(S_k) and Update R(S)
 Update D^- : D^- ← D^- − S_k
End While
Output: S = {S_1, S_2, ..., S_N, R(S)
```

For both types of rounds, the GH calculates the coverage gain g_j, as defined in Section III.B, of all the deployment points $d_j \in \mathcal{N}_{next}$ and adds the point with the *highest* gain, denoted by g_{max}, to S_k. In the case where more than one deployment point have the maximum gain or if none of the deployment points belonging to $\mathcal{N}_{next}$ have a non-zero coverage gain, the GH chooses a deployment point from $\mathcal{N}_{next}$ randomly. The GH terminates when the combined reliability of the constructed connected covers meets the reliability constraint, i.e. $R(\mathcal{S}) \geq R_{min}$.

Tables 3, 4 and 5 summarize the results obtained from applying the proposed ACO algorithm and the GH described above to the eight test cases in Table 1 at $R_{min} = 0.99, 0.999$ and 0.9999 respectively. The tables show the lowest ('Best'), highest ('Worst') and the average ('Avg.') total number of deployment points in the connected covers constituting the solutions obtained from both methods in ten independent runs. The tables also show the success rate ('SR') in percentage of each method in finding a solution to each MCRC-SDP instance that fulfills all the constraints of the problem, i.e. a feasible solution. Since fulfilling the problem constraints expressed in (4), (5) and (6) is guaranteed by the solution construction procedure followed by both methods, SR is actually the success rate of each method in finding solutions

TABLE 3. Comparison between the proposed ACO algorithm and the greedy heuristic on the test cases at $R_{min} = 0.99$.

Test Case	Proposed ACO algorithm				Greedy Heuristic			
	Best	Worst	Avg.	SR (%)	Best	Worst	Avg.	SR (%)
TC1	**8**	**8**	**8**	**100**	10	11	10.3	20
TC2	**8**	**8**	**8**	**100**	10	12	11	50
TC3	**10**	**10**	**10**	**100**	11	21	13.9	40
TC4	**10**	**10**	**10**	**100**	12	19	17	70
TC5	**18**	**18**	**18**	**100**	19	21	20.2	30
TC6	**18**	**19**	**18.2**	**100**	21	24	22.3	30
TC7	**18**	**20**	**19.2**	**100**	22	24	22.7	50
TC8	**11**	**11**	**11**	**100**	22	24	23.3	0

TABLE 4. Comparison between the proposed ACO algorithm and the greedy heuristic on the test cases at $R_{min} = 0.999$.

Test Case	Proposed ACO algorithm				Greedy Heuristic			
	Best	Worst	Avg.	SR (%)	Best	Worst	Avg.	SR (%)
TC1	**13**	**13**	**13**	**100**	15	16	15.9	10
TC2	**13**	**13**	**13**	**100**	14	18	16.1	30
TC3	**16**	**16**	**16**	**100**	18	29	22.2	0
TC4	**16**	**17**	**16.3**	**100**	18	27	25	30
TC5	**24**	**24**	**24**	**100**	27	29	27.9	30
TC6	**24**	**26**	**25**	**100**	28	32	29.7	20
TC7	**26**	**27**	**26.5**	**100**	30	34	31.3	30
TC8	**17**	**17**	**17**	**100**	30	32	30.6	0

TABLE 5. Comparison between the proposed ACO algorithm and the greedy heuristic on the test cases at $R_{min} = 0.9999$.

Test Case	Proposed ACO algorithm				Greedy Heuristic			
	Best	Worst	Avg.	SR (%)	Best	Worst	Avg.	SR (%)
TC1	**18**	**18**	**18**	**100**	21	22	21.5	0
TC2	**18**	**18**	**18**	**100**	21	26	21.8	20
TC3	**22**	**23**	**22.5**	**100**	30	38	35.8	0
TC4	**22**	**23**	**22.2**	**100**	24	36	30.8	0
TC5	**30**	**31**	**30.9**	**100**	34	38	35.7	10
TC6	**32**	**33**	**32.3**	**100**	36	39	37.1	0
TC7	**33**	**34**	**33.7**	**100**	37	41	39	10
TC8	**31**	**33**	**31.9**	**100**	37	41	39.1	0

that fulfill the redundancy constraint expressed in (7) as well as the other constraints. For each problem instance, the best results between both methods are written in bold.

From the results shown in Tables 3, 4 and 5, it can be observed that the proposed ACO algorithm has a success rate of 100% for all the twenty four problem instances under consideration, whereas the success rate of the GH does not exceed 70% and is on average considerably lower than 70%. For all the problem instances, the GH was capable of finding solutions that satisfy the MCRC-SDP constraints expressed in (4)-(6), i.e. it was capable of constructing solutions consisting of non-overlapping connected covers with a combined reliability greater than or equals the specified minimum value R_{min}. However, it failed in a considerable number of runs in obtaining solutions that satisfy the redundancy constraint, i.e. solutions that consist *only* of minimal connected covers.

This implies that the GH can consistently obtain feasible solutions to the *relaxed* version of the MCRSDP problem, which has the same objective function and constraints as the original version but excluding the redundancy constraint. The failure of the GH in consistently constructing feasible solutions to the original, more restrictive version of the MCRC-SDP can be attributed to the 'greedy' method if follows in constructing connected covers, which aims at reducing the chance of adding redundant deployment points to connected covers but fails to eliminate it completely. A redundant deployment point or points can be added to a given connected cover $\boldsymbol{S}_k$ in the following case. At any given stage in constructing $\boldsymbol{S}_k$, all the deployment points belonging to the set $\mathcal{N}_{next}$ happen to have a zero coverage gain. In this case, the GH selects a deployment point at random from $\mathcal{N}_{next}$ to maintain connectivity of $\boldsymbol{S}_k$. Hence, the selected deployment point is redundant to $\boldsymbol{S}_k$ in terms of coverage but non-redundant in terms connectivity thus far. However, depending on the deployment points selected by the GH in the following rounds till the completion of $\boldsymbol{S}_k$, this redundant deployment point(s) in terms of coverage may become redundant in terms of connectivity as well, meaning that its elimination from $\boldsymbol{S}_k$ would not compromise its coverage or connectivity. Hence, such a deployment point(s) becomes fully redundant and consequently $\boldsymbol{S}_k$ becomes a non-minimal connected cover.

On the other hand, the results show that the proposed ACO algorithm is consistently capable of finding feasible solutions to the MCRC-SDP with a success rate of 100% over all tested problem instances. This is attributed to the design of the ACO algorithm's cost function expressed in (15). As discussed earlier, the ACO solution/tour construction procedure is similar to GH in terms of the underlying basic idea and heuristic information used in deployment points' selection. Hence, some ants may construct infeasible tours due to the violation of the redundancy constraint if one or more of the connected covers in the corresponding solutions are non-minimal. However, the cost function penalizes these infeasible tours and consequently as the algorithm progresses, the pheromone trail levels (which are updated at the end of each iteration using (16)) will reinforce feasible tours and increase the proportion of ants which construct feasible solutions over that of ants which construct non-feasible ones. This process is accelerated by the use of the proposed LS procedure, which converts non-feasible tours into feasible ones by eliminating redundant deployment points from connected covers in the case where this elimination would not lead to the violation of the reliability constraint, i.e. the reduction of combined reliability $R(\mathbb{S})$ below R_{min}.

It can also be observed from the tabulated results that for each of the eight test cases, the success rate of the GH declines as the value of R_{min} increases. This behaviour is expected since the number of connected covers required to satisfy the reliability constraint increases with the increase of the value of R_{min}. As the number of connected covers the GH has to construct to meet R_{min} increases, the probability that a non-minimal connected cover is constructed increases as well. Consequently, this increases the probability that the GH obtains a non-feasible solution with one or more non-minimal connected covers which constitutes a failure.

Results also show that the quality of the obtained solutions by the proposed ACO algorithm is superior to that of the solutions obtained by the GH by more than 20% on average. In all the problem instances, the highest total number of deployment points in the solutions (i.e. 'Worst' solution) obtained by the ACO algorithm is lower than the lowest total number of deployment points in the solutions (i.e. 'Best' solution) obtained by the GH. This implies that even for the problem instances where the GH succeeded in obtaining solutions consisting only of minimal covers (i.e. feasible solutions) with a success rate higher than null, the proposed ACO algorithm was capable of finding feasible solutions with significantly higher quality, i.e. solutions of a lower deployment cost. This is attributed to the search efficiency of the proposed ACO algorithm, which is capable of finding solutions that consist of minimal connected covers of smaller sizes than the minimal connected covers in the feasible solutions obtained by the GH. Furthermore, the ACO solutions have higher combined reliability levels than those of the GH solutions. This is because the reliability of a minimal connected cover is inversely proportional with its size, since reliability in this case is equal to the probability of the single event that all the deployed SNs are 'on'. This probability increases when there are fewer deployed SNs, i.e. when there are fewer deployment points in the minimal connected cover.

Test cases TC3, TC4 and TC8 show a greater advantage of the proposed ACO algorithm over the GH in terms of solution quality as compared to the rest of the test cases at the three considered levels for R_{min}. This can be attributed to the following. In these problem instances, the GH obtained solutions that consisted of an entire additional connected cover when compared to the solutions obtained by the ACO algorithm. This is because the GH constructs one or more minimal connected covers of non-optimal size (i.e. with lower reliability) in its earlier rounds, i.e. at the beginning of constructing a solution to the problem. This has caused the GH to have to construct an additional connected cover to meet the reliability constraint. This situation does not occur in the solutions obtained by the ACO due to its search efficiency.

V. CONCLUSION

In this paper, we considered the problem of deploying a WSN that meets a specified minimum level of reliability during its mission time at a minimum network deployment cost. To minimize the internal interference, bandwidth usage and energy consumption throughout the network's mission time, we defined the problem as the problem of finding a number of non-overlapping minimal connected covers of the targeted region of interest such that the combined reliability level of these connected covers meets or exceeds the specified minimum level of reliability at a minimum deployment cost. We coined this problem the Minimum Cost Reliability Constrained Sensor Node Deployment Problem (MCRC-SDP). We proved that the MCRC-SDP is NP-complete and proposed an ACO-based approach, coupled with a local search procedure, to solve it. Our experimental results on twenty four problem instances with different operational parameters demonstrated the effectiveness of the proposed approach in finding high-quality solutions to the problem. Results also show that the quality of the obtained solutions by the proposed ACO algorithm is superior to that of the solutions obtained by a Greedy Heuristic by more than 20% on average.

REFERENCES

[1] A. Flammini and E. Sisinni, "Wireless sensor networking in the Internet of Things and cloud computing era," in *Proc. Eur. Conf. Solid-State Transducers (EUROSENSOR)*, vol. 87. 2014, pp. 672–679.

[2] P. Harrop and R. Das, "Wireless sensor networks (WSN) 2014-2024: Forecasts, Technologies, Players," IDTechEx, Cambridge, U.K., Tech. Rep. 1.1-1.10, 2014. [Online]. Availanbe: http://www.idtechex.com/research/reports/wireless-sensor-networks-wsn-2014-2024-forecasts-technologies-players-000382.de.asp?viewopt=orderinfo

[3] L. Lei, Y. Kuang, X. Shen, K. Yang, J. Qiao, and Z. Zhong, "Optimal reliability in energy harvesting industrial wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5399–5413, Aug. 2016.

[4] F. Koushanfar, M. Potkonjak and A. Sangiovanni-Vecentelli, "Fault tolerance in wireless sensor networks," in *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, 1st ed. Boca Raton, FL, USA: CRC Press, 2005, ch. 36.

[5] D. S. Deif and Y. Gadallah, "Classification of wireless sensor networks deployment techniques," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 834–854, 2nd Quart., 2014.

[6] W. Ke, B. H. Liu, and M. J. Tsai, "Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points Is NP-complete," *IEEE Trans. Comput.*, vol. 56, no. 5, pp. 710–715, May 2007.

[7] Y. Xu and X. Yao, "A GA approach to the optimal placement of sensors in wireless sensor networks with obstacles and preferences," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jun. 2006, pp. 127–131.

[8] Q. Wu, N. Rao, X. Du, S. Iyengar, and V. Vaishnavi, "On efficient deployment of sensors on planar grid," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2721–2734, 2007.

[9] B. Carter and R. Ragade, "An extensible model for the deployment of non-isotropic sensors," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Feb. 2008, pp. 22–25.

[10] B. Carter and R. Ragade, "A probabilistic model for the deployment of sensors," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Feb. 2009, pp. 7–12.

[11] D. S. Deif and Y. Gadallah, "Wireless sensor network deployment using a variable-length genetic algorithm," in *Proc. Wireless Commun. Netw. Conf. (WCNC)*, 2014, pp. 2450–2455.

[12] D. Li, W. Liu, and L. Cui, "EasiDesign: An improved ant colony algorithm for sensor deployment in real sensor network system," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2010, pp. 1–5.

[13] X. Liu, "Sensor deployment of wireless sensor networks based on ant colony optimization with three classes of ant transitions," *IEEE Commun. Lett.*, vol. 16, no. 10, pp. 1604–1607, Oct. 2012.

[14] H. Kim and S. W. Han, "An efficient sensor deployment scheme for large-scale wireless sensor networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 98–101, Jan. 2015.

[15] G. Huang, D. Chen, and X. Liu, "A node deployment strategy for blindness avoiding in wireless sensor networks," *IEEE Commun. Lett.*, vol. 19, no. 6, pp. 1005–1008, Jun. 2015.

[16] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2364–2376, Oct. 2015.

[17] K. Sakai, M. T. Sun, W. S. Ku, T. H. Lai, and A. V. Vasilakos, "A framework for the optimal k-Coverage deployment patterns of wireless sensors," *IEEE Sensors J.*, vol. 15, no. 12, pp. 7273–7283, Dec. 2015.

[18] S. K. Gupta, P. Kuila, and P. K. Gana, "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks," *Comput. Elect. Eng.*, vol. 56, pp. 544–556, Nov. 2016.

[19] H. M. F. AboElFotoh, S. S. Iyengar, and K. Chakrabarty, "Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures," *IEEE Trans. Rel.*, vol. 54, no. 1, pp. 145–155, Mar. 2005.

[20] E. I. Gokce, A. K. Shrivastava, and Y. Ding, "Fault tolerance analysis of surveillance sensor systems," *IEEE Trans. Rel.*, vol. 62, no. 2, pp. 478–489, Jun. 2013.

[21] D. S. Deif and Y. Gadallah, "Reliability assessment of wireless sensor network deployments," in *Proc. IFIP Wireless Days (WD)*, 2016, pp. 1–6.

[22] I. B. Gertsbakh and Y. Shpungin, *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*, 1st ed. CRC Press, 2010, pp. 17–23.

[23] J. Levine and F. Ducatelle, "Ant colony optimization and local search for bin packing and cutting stock problems," *J. Oper. Res. Soc.*, vol. 55, no. 7, pp. 705–716, 2004.

[24] F. Neumann, D. Sudholt, and C. Witt, "Rigorous analyses for the combination of ant colony optimization and local search," in *Proc. 6th Int. Conf. Ant Colony Optim. Swarm Intell.*, 2008, pp. 132–143.

[25] Y. Lin, J. Zhang, H. S.-H. Chung, W. H. Ip, Y. Li, and Y.-H. Shi, "An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 3, pp. 408–420, May 2012.

[26] W. Kuo and M. J. Zuo, *Optimal Reliability Modeling: Principles and Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2003, pp. 85–102.

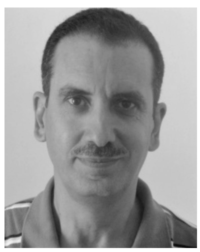
[27] J. Virkki, Y. Zhu, Y. Meng, and L. Chen, "Reliability of WSN hardware," *Int. J. Embedded Syst.*, vol. 1, no. 2, pp. 1–10, 2011.

[28] N. Baccour *et al.*, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Trans. Sensor Netw.*, vol. 8, no. 4, p. 34, 2012.

[29] Y. Zou and K. Chakrabarty, "A distributed coverage-and connectivity-centric technique for selecting active nodes in wireless sensor networks," *IEEE Trans. Comput.*, vol. 54, no. 8, pp. 978–991, Aug. 2005.

[30] M. Dorigo and T. Stutzle, *Ant Colony Optmization*. Cambridge, MA, USA: MIT Press, 2004, pp. 25–100.

DINA S. DEIF (S'12) received the B.Sc. and M.Sc. in electronics and telecommunication engineering from Ain-Shams University, Cairo, Egypt, in 2006 and 2011, respectively. She is currently pursuing the Ph.D. degree with the Department of Electronics Engineering, The American University in Cairo, Egypt. Her current research interests are in the field of wireless sensor networks, primarily in deployment optimization and fault tolerance. From 2006 to 2011, she was a Teaching Assistant with the Department of Electronics Engineering, Misr International University in Cairo, Egypt.



Yasser Gadallah (S'91–M'05–SM'09) received the M.Sc. degree from Queens University, Kingston, ON, Canada, and the Ph.D. degree from Carleton University, Ottawa, ON, Canada, both in electrical engineering. He is currently an Associate Professor of Communication Networks with the American University in Cairo (AUC), Cairo, Egypt. Prior to joining the AUC, he was an Associate Professor with the Electronics and Communications Engineering Department, Misr International University, Cairo, Egypt. Earlier, he was an Assistant Professor of Networking Systems with the College of Information Technology, UAE University. Prior to joining the UAE University, He was a Research Scientist with the Communications Research Centre (CRC), Ottawa, Canada, in the area of wireless sensor networks. He has also worked for many years in the networking and telecommunications industry. He worked as a Software Development Manager with Cisco Systems for several years, where he was involved in numerous R&D projects, which resulted in the introduction of several high profile Cisco products. He also worked with Nortel Networks as a Software Project Leader and prior to that as a Software Designer. His research interests include the Internet of Things issues, machine to machine communications issues, wireless sensor networks, mobile ad hoc networks, and broadband wireless access networks. His biography appears in Marquis Who's Who in the World in 2010. He has served as an Organizer, a technical program committee member, and a Reviewer for several international conferences and journals. He has several technical and scientific publications in the networking, communications, and IT fields.