Node Placement and Path Planning for Improved Area Coverage in Mixed Wireless Sensor Networks

Survi Kumari and Seshan Srirangarajan, *Member, IEEE*

***Abstract*—For the large-scale monitoring of a physical phenomena using a wireless sensor network (WSN), a large number of static and/or mobile sensor nodes are required, resulting in higher deployment cost. In this work, we develop an efficient algorithm that can employ a small number of static nodes together with a set of mobile nodes for improved area coverage. An efficient deployment of static nodes and guided mobility of the mobile nodes is critical for maximizing the area coverage. To this end, we propose three mixed integer linear programming (MILP) formulations. The first formulation efficiently deploys a set of static nodes and the other two formulations plan the path of a set of mobile nodes so as to maximize the area coverage and minimize the total number of movements required to achieve the desired coverage. We present extensive performance evaluation of the proposed algorithms and their comparison with benchmark approaches. The simulation results demonstrate the superior performance of the proposed algorithms for different network area and number of static and mobile nodes.**

***Index Terms*—Area coverage, mixed wireless sensor network, mixed-integer linear programming (MILP), mobile nodes.**

I. Introduction

Wireless sensor networks (WSN) are being used in various monitoring or surveillance applications and ensuring full area coverage is one of the key objectives in such applications [1]. Deployment of only static nodes typically leads to coverage holes and overlapping coverage due to sub-optimal placement of nodes and/or nodes becoming non-functional over a period of time after the initial deployment. Increasing the number of nodes or their sensing region are typically not cost effective. In addition, these measures do not address the issue of non-functional nodes or overlapping coverage or the effect of environmental factors on the network. Thus, a mixed WSN, a network with a combination of static and mobile nodes, has been proposed to address these limitations [2].

Mobility in sensor nodes is the ability of the nodes to move and change their locations post their initial deployment. With the advancement in technology, the applications of mobile sensor nodes have gradually increased. The use of mobile nodes significantly improves the possibility of maintaining a robust network coverage [3]. In WSN literature and applications, the mobility of nodes is primarily exploited in two ways. In some cases, the fusion center or sink is mobile and the mobile sink moves throughout the network area to gather data from the sensing nodes [4], [5]. In other cases, one or more of the sensing nodes are mobile and move through the network area to record data along with location information and deliver the collected data to the fusion center.

A WSN is usually deployed in an area of interest for monitoring or detection applications. The ability to monitor the entire area of interest at all time instances is referred to as continuous coverage. However, in many applications, periodic monitoring is sufficient instead of continuous monitoring, and this is referred to as sweep coverage. Typical examples include data gathering and message ferrying [6], [7]. Sweep coverage involves mobile nodes. In a mixed WSN, efficient path-planning strategies need to be developed for the mobile nodes to achieve optimal coverage. In a mixed WSN, there are static nodes in addition to mobile nodes that need to be optimally placed.

A. Related Work

Coverage path planning is an active area of research with the growing popularity of mobile nodes. Several works utilize the mobility of nodes in order to remedy coverage issues that occur post the deployment of static nodes. The method in [8] employs the harmony search metaheuristic algorithm to identify the minimum number of mobile sensors required for monitoring the regions inaccessible to randomly deployed static sensors. In [9], the coverage holes are assumed to be irregular polygons, referred to as blind zones. The optimal placement of mobile sensors is determined, based on the locations of blind zones, to minimize redundant coverage. Similarly, [10] presented a genetic algorithm-based solution for deploying the minimum number of mobile sensors in order to address coverage holes efficiently.

Using mobility merely for node relocation is an underutilization of the potential of mobile sensors. Area coverage with the help of mobile relays, where mobile nodes gather data from static sensors and carry them to the sink, has been studied in [11]. Authors in [12] consider network lifetime maximization using a mobile sink. They maximize the total sojourn time of a single mobile node, acting as a sink, using a linear programming model. Although sink mobility could improve network lifetime compared to mobile relays, frequent broadcasting of sink position leads to significant overhead. A review of several other

Manuscript received 6 January 2024; accepted 28 May 2024. Date of publication 11 June 2024; date of current version 18 June 2024. This letter was recommended for publication by Associate Editor K. Tiwari and Editor A. Bera upon evaluation of the reviewers' comments. *(Corresponding author: Seshan Srirangarajan.)*

Survi Kumari is with the Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, Delhi 110016, India (e-mail: survikumari7@gmail.com).

Seshan Srirangarajan is with the Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, Delhi 110016, India, and also with the Bharti School of Telecommunication Technology and Management, Indian Institute of Technology Delhi, New Delhi, Delhi 110016, India (e-mail: seshan@ee.iitd.ac.in).

Digital Object Identifier 10.1109/LRA.2024.3412590

methods that exploit mobility of nodes for network coverage problems can be found in [13], [14].

Mobile nodes have also been used for sweep coverage. The authors in [15] presented approximation algorithms that minimize the time taken by mobile nodes for visiting a set of targets. The objective is to reduce the target detection period, while minimizing the trajectory length of the mobile nodes. In [16], the static sensors detect coverage holes by using Voronoi diagrams and bid for mobile sensors based on the size of the coverage hole detected by them. Mobile sensors accept the highest bids and move to heal the largest holes. To reduce the movement distance of mobile nodes, a proxy-based bidding protocol is employed where mobile sensors perform virtual movements from small holes to large holes and only perform physical movements after the final destinations are identified.

In [2], the authors consider various cost measures to address trade-offs between area coverage and distance traveled by mobile nodes or information exchange among nodes. In [17], the authors proposed a distributed technique for iteratively computing the paths for mobile nodes in a greedy manner. They employ a bidding approach similar to that in [16] while using the movement scheme described in [2], [18]. In addition, the static nodes employ the zoom algorithm to determine the largest hole. The zoom algorithm divides the neighborhood into four equal regions and chooses the region with the largest uncovered area. Further details about the zoom algorithm can be found in [19], [20].

The authors in [21] plan the path of a mobile node to maximize area coverage while traversing through the uncovered regions. They present three mixed integer linear programming (MILP)-based formulations for this path planning exercise. A similar recent work [22] is aimed at identifying the shortest tour for a mobile sensor to maintain the desired coverage level in a smart grid application. A mixed integer programming model, free from restrictive mobility pattern assumptions, is presented to determine the shortest trajectory, and heuristic-based algorithms are presented as the solution methodology.

Our work is inspired by [21], which proposes an optimization framework to plan the path of a single mobile node to maximize area coverage and/or minimize the total trip time for the mobile node. However, relying on a single mobile node for path planning can result in coverage limitations, vulnerability to failure, and reduced efficiency. Moreover, in [21], the static nodes are deployed randomly which may lead to undesired effects on the coverage performance. In our work, we propose a placement strategy for a set of static nodes for maximizing their coverage while prioritizing the coverage of the boundary areas of the network. Next, a small set of mobile nodes are employed to traverse through and monitor the uncovered regions in an efficient manner. In this process, the mobile nodes tend to visit some locations multiple times due to traveling range constraints. Thus, in the proposed path planning strategy, we introduce an additional constraint that limits the overlapping coverage. In our work, we use MILP-based formulations as these offer advantages in terms of their optimality guarantees. They can also incorporate real-world constraints effectively and produce consistent results.

B. Our Contribution

In this work, we present strategies to plan the path of a set of mobile nodes to achieve sweep coverage through periodic monitoring of the network area. The proposed system model assumes a small number of mobile sensor nodes along with a limited number of static sensor nodes. The objective is to plan the paths of the mobile nodes, primarily over the uncovered areas, so as to improve the overall area coverage and total trip time. We also propose an algorithm for the deployment of static nodes that aids in the path planning of the mobile nodes by avoiding the challenges that can arise due to a random deployment of static nodes, such as boundary coverage holes [23], network partition, and redundant coverage [18]. The proposed system model includes a parameter that determines the permissible level of redundant coverage or overlapping coverage during the path planning process. The proposed strategies are formulated as MILPs. The key contributions of our work are as follows:

- An MILP-Static formulation that places static nodes to maximize area coverage while giving more weight to covering the network boundary areas.
- An MILP-Cov formulation that plans the path of a set of mobile nodes while maximizing the coverage area within a given number of movements or time steps.
- An MILP-Mov formulation that plans the path of a set of mobile nodes while minimizing the number of movements required to achieve a desired coverage level.

II. System Model

Consider a rectangular network area which is subdivided into grids with $M \times N$ unit square cells. Each grid cell location is denoted by coordinates (i, j) with $1 \leq i \leq M$ and $1 \leq j \leq N$. We assume a mixed WSN, consisting of a few static and mobile nodes for monitoring the network area, and a single data sink node. The first objective is to place the static nodes within the network area in a manner that aids the path planning of the mobile nodes. Next, we plan the paths of the mobile nodes to maximize the area coverage and minimize the trip time. The goal is to place the static nodes and plan the paths of mobile nodes such that all the grid cells within the network area are covered at least once (also called 1-*coverage*) by a sensor node (static or mobile). The notations of system parameters and decision variables used in the proposed system model are listed in Table I, which are described next.

Let N_s and N_m denote the number of static and mobile nodes, respectively. The coordinate of the grid cell in which a sensor is located is defined as the location of that sensor. Let r_s denote the sensing radius (or sensing range) of static/mobile nodes such that r_s is the number of cells that can be sensed by a node in each direction from its current location. The network is assumed to be connected such that the sensor nodes can communicate with the sink node at all times.

After the initial deployment of static nodes is completed, the paths to be followed by the mobile nodes is planned centrally. The path planning is an iterative procedure where the location of each mobile node is computed simultaneously at each time-step. The mobile nodes sense the parameter(s) of interest in the cells within their sensing range (r_s) and then move to their next locations within the one-step traveling range, denoted by ρ_x and ρ_y along the x and y directions, respectively. The one-step traveling range of a mobile node is the maximum number of cells up to which the mobile node can move in each direction from its current location in one time-step. The paths of mobile nodes are computed in advance (or offline) and is subject to revision

TABLE I
NOTATIONS

Symbol	Definition
N_s	Number of static sensor nodes
N_m	Number of mobile sensor nodes
$\mathcal{C}$, $\lvert\mathcal{C}\rvert$	Set of all the cells in the network area and its cardinality
$\mathcal{B}$	Set of cells at the boundary of the network area
$\mathcal{A}$	Set of cells other than the boundary cells
$\mathcal{C}_s$	Set of cells covered by static nodes
$\overline{\mathcal{C}}_s$	Set of cells not covered by static nodes
cr	Area coverage ratio
r_s	Sensing radius of a sensor node
ρ_x, ρ_y	One-step traveling range of mobile node along x and y directions
$\hat{s}_o$, $\hat{m}_o$	Overlapping cell coverage factor for static and mobile sensor nodes' coverage
$K_{\max}$	Maximum number of time-steps
$x^s_{i,j}$	$x^s_{i,j} = 1$ if the s^{th} static sensor node is located at cell (i, j), otherwise $x^s_{i,j} = 0$.
$x^{l,k}_{i,j}$	$x^{l,k}_{i,j} = 1$ if the l^{th} mobile node is located at cell (i, j) in the k^{th} time-step.
$c^s_{i,j}$	$c^s_{i,j} = 1$ if the cell (i, j) is covered by the s^{th} static sensor node, otherwise $c^s_{i,j} = 0$.
$c^{l,k}_{i,j}$	$c^{l,k}_{i,j} = 1$ if the cell (i, j) is covered by the l^{th} mobile node in the k^{th} time-step, otherwise $c^{l,k}_{i,j} = 0$.
$c_{i,j}$	$c_{i,j} = 1$ if the cell (i, j) is covered by a sensor node during any time-step, otherwise $c_{i,j} = 0$.

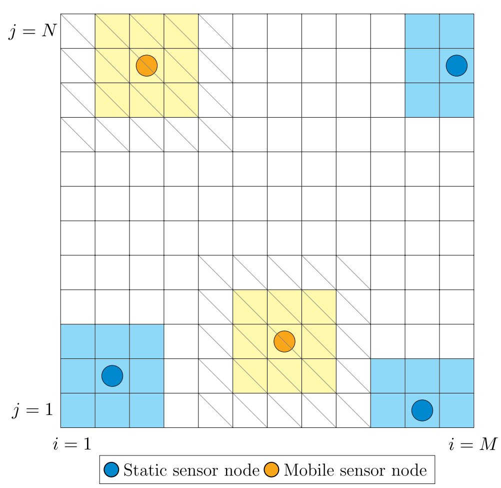


Fig. 1. A typical network model. The colored cells around a sensor node indicate the sensing region of that sensor node ($r_s = 1$). The cells marked with diagonal lines around each mobile node represent potential locations where the mobile node could move to in the next time-step ($\rho_x = \rho_y = 2$).

only in the event of changes to the physical structure of the area of interest.

An example network model is shown in Fig. 1. The various parameter values for this network are $M = 12$, $N = 12$, $N_s = 3$, $N_m = 2$, $r_s = 1$, $\rho_x = \rho_y = 2$, $\mathcal{C} = \{(i, j)|i = 1,\ldots, M; j = 1,\ldots, N\}$, $|\mathcal{C}| = 144$, and $\mathcal{B} = \{(i = 1, j = 1,\ldots, N), (i = M, j = 1,\ldots, N), (i = 1,\ldots, M, j = 1), (i = 1,\ldots, M, j = N)\}$. This is a 12×12 grid network with 3 static nodes providing coverage for 21 cells (which form $\mathcal{C}_s$). The cells not covered by static nodes ($\overline{\mathcal{C}}_s$) are to be covered using 2 mobile nodes with a one-step traveling range of two cells along each direction.

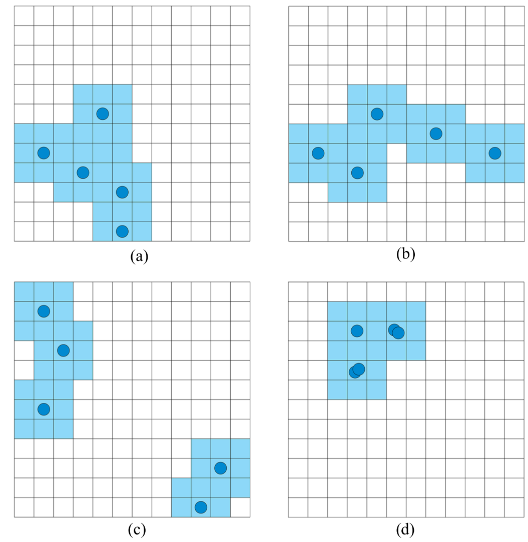


Fig. 2. Possible challenges due to random deployment of static sensor nodes in a network area. (a), (b) network partition, (c) boundary coverage hole, and (d) overlapping/redundant coverage.

III. PROPOSED STRATEGIES

In this section, we describe the proposed algorithms for the placement of static nodes and path planning of the mobile nodes. Consider the variables $x^s_{i,j}$ and $x^{l,k}_{i,j}$ which indicate the locations of static and mobile nodes, respectively. Also, let $c^s_{i,j}$, $c^{l,k}_{i,j}$, and $c_{i,j}$ be the coverage variables. These variables are defined in Table I. We define $x^s_{i,j}$ and $x^{l,k}_{i,j}$ as binary variables, whereas $c_{i,j}$, $c^s_{i,j}$, and $c^{l,k}_{i,j}$ are defined as continuous variables in the range $[0, 1]$. Thus, the proposed node placement and path planning strategies result in MILP formulations which are described next.

It is to be noted that even though $c_{i,j}$, $c^s_{i,j}$, and $c^{l,k}_{i,j}$ have been defined as continuous variables, due to their defined range and other constraints, they will behave as binary variables. This is because the constraints have been formulated such that they can only take values 0 or 1. They have been defined as continuous variables in order to reduce the complexity of the integer linear programs (ILPs). It is well-known that the complexity of ILPs increase exponentially with the number of binary/integer variables [21].

A. Static Node Placement

The first step is to place the static sensor nodes within the network area so as to maximize the area coverage. As noted earlier, the random deployment of static nodes can result in challenges such as boundary coverage holes, network partitioning, and redundant coverage. Some of these scenarios are illustrated in Fig. 2. In addition, if the static nodes do not cover the boundary cells, the mobile nodes would have to cover these resulting in a larger number of movements and redundant coverage. Thus, in the proposed static node placement strategy we would like to maximize the covered area while giving more importance/weight to covering the cells at the network boundary. The static node placement strategy can be formulated as the following optimization problem, referred to as MILP-Static.

$$\max \left(\sum_{\substack{s=1 \\ (i,j)\in\mathcal{A}}}^{N_s} c^s_{i,j} + \alpha \sum_{\substack{s=1 \\ (i,j)\in\mathcal{B}}}^{N_s} c^s_{i,j} \right) \tag{1}$$

$$\text{s.t.} \sum_{(i,j)\in\mathcal{C}} x^s_{i,j} = 1; \quad s = 1,\ldots,N_s \tag{2}$$

$$c^s_{i,j} = \sum_{p=-r_s}^{r_s} \sum_{q=-r_s}^{r_s} x^s_{i+p,j+q}; \forall (i,j) \in \mathcal{C}; s = 1,\ldots,N_s \tag{3}$$

$$\sum_{s=1}^{N_s} c^s_{i,j} \leq \hat{s}_o; \quad \forall (i,j) \in \mathcal{C} \tag{4}$$

Objective function: The variable $c^s_{i,j}$ represents the status of cell (i,j) being covered/not covered by the static node s. The sum of $c^s_{i,j}$ over all s and (i,j) represents the total number of cells covered by static nodes. In order to give different weight to the coverage of boundary cells, we define the set of boundary cells ($\mathcal{B}$) as the single row/column of cells along the edge of the network area with $|\mathcal{B}| = 2(M + N - 2)$. The objective function in (1) maximizes the total number of covered cells, where α is a weight parameter associated with the coverage of the boundary cells. Using $\alpha > 1$, gives more importance to the coverage of boundary cells. In our simulations, we set $\alpha = 4$.

Position constraint: In order to ensure that each sensor node is placed at only one grid cell, the position variable $x^s_{i,j}$ for each s over all the grid cells must sum up exactly to 1. In simpler terms, it guarantees that one sensor node cannot occupy multiple grid cells simultaneously. This can be represented mathematically as in (2).

Coverage constraint: Each sensor node can sense r_s number of cells in each direction along with the one in which it is currently placed/present. Therefore a cell (i,j) is considered to be covered by a sensor node only if that node is placed/present in any of the cells that are within r_s from (i,j). This can be represented mathematically by the equality constraint (3). This ensures that $c^s_{i,j} = 1$, if the cell (i,j) is within the sensing range of the s^{th} static node.

Overlapping coverage constraint: To ensure efficient utilization of the nodes' resources, there is a need to avoid scenarios where multiple nodes are placed at the same grid cell, or where a nodes' coverage region excessively overlaps with the coverage region of other nodes. To achieve this, we limit the overlap of cell coverage among the nodes. This is accomplished by ensuring that any given cell (i,j) is covered by at most a pre-defined number of times or nodes, as determined by the overlapping cell coverage factor which is denoted by $\hat{s}_o$ in the case of static nodes. This constraint is mathematically expressed as (4).

B. Coverage Maximization

After the static nodes have been placed (using MILP-Static), the mobile nodes must traverse the network area to cover the uncovered cells $(i,j) \in \overline{\mathcal{C}}_s$, so as to maximize the area coverage. The path planning of the mobile nodes, to maximize area coverage, can be formulated as the following optimization problem and is referred to as MILP-Cov.

$$\max \sum_{(i,j)\in\overline{\mathcal{C}}_s} c_{i,j} \tag{5}$$

$$\text{s.t.} \sum_{(i,j)\in\overline{\mathcal{C}}_s} x^{l,k}_{i,j} = 1; \quad l = 1,\ldots,N_m; k = 1,\ldots,K_{\max} \tag{6}$$

$$x^{l,k+1}_{i,j} = \sum_{p=-\rho_x}^{\rho_x} \sum_{q=-\rho_y}^{\rho_y} x^{l,k}_{i+p,j+q}; \quad \forall (i,j) \in \overline{\mathcal{C}}_s; \quad l = 1,\ldots,N_m; k = 1,\ldots,(K_{\max} - 1) \tag{7}$$

$$c^{l,k}_{i,j} = \sum_{p=-r_s}^{r_s} \sum_{q=-r_s}^{r_s} x^{l,k}_{i+p,j+q}; \quad \forall (i,j) \in \overline{\mathcal{C}}_s; \quad l = 1,\ldots,N_m; k = 1,\ldots,K_{\max} \tag{8}$$

$$c_{i,j} \geq c^{l,k}_{i,j}; \forall (i,j) \in \overline{\mathcal{C}}_s; l = 1,\ldots,N_m; k = 1,\ldots,K_{\max} \tag{9}$$

$$c_{i,j} \leq \sum_{l=1}^{N_m} \sum_{k=1}^{K_{\max}} c^{l,k}_{i,j}; \quad \forall (i,j) \in \overline{\mathcal{C}}_s \tag{10}$$

$$\sum_{l=1}^{N_m} \sum_{k=1}^{K_{\max}} c^{l,k}_{i,j} \leq \hat{m}_o; \quad \forall (i,j) \in \overline{\mathcal{C}}_s \tag{11}$$

$$0 \leq c_{i,j}, c^{l,k}_{i,j} \leq 1; \forall (i,j) \in \overline{\mathcal{C}}_s; l = 1,\ldots,N_m; \quad k = 1,\ldots,K_{\max} \tag{12}$$

Objective function: The coverage variable $c_{i,j}$ denotes whether the grid cell (i,j) has been sensed/covered by any sensor node or not. Our aim here is to maximize the number of grid cells that are covered by the mobile nodes. The objective function (5), thus, involves a sum of the coverage variable values $c_{i,j}$ over the set of uncovered grid cells $\overline{\mathcal{C}}_s$.

Position constraint: Physically each mobile node can occupy only one grid cell at any given time-step. This can be ensured if the position variable $x^{l,k}_{i,j}$ for each mobile node summed across all the uncovered cells is exactly equal to 1. It is similar to the position constraint of MILP-Static, with the distinction that the nodes are now mobile and this condition must be satisfied at each time-step. This constraint can be expressed mathematically as in (6).

Mobility constraint: The mobile nodes need to traverse through the uncovered cells to cover them periodically, however, their movement range is limited. We assume that a mobile node can move only once in a time-step, therefore, if it is at a grid cell (i,j) in a time-step, then it must have been at one of the cells within its one-step traveling range in the previous time-step. This is expressed as constraint (7). Although the one-step traveling ranges ρ_x and ρ_y along x and y directions, respectively, can have different values, we assume them to be equal in our simulations for simplicity.

Coverage constraints: To define coverage constraints in this formulation, we map the coverage achieved by each mobile node onto the global coverage variable $c_{i,j}$. A variable $c^{l,k}_{i,j}$ is defined to account for grid cells covered by each mobile node individually at each time-step. Constraint (8) ensures that a cell (i,j) is considered as covered, by the l^{th} mobile node at the k^{th}

time-step, if the grid cell (i, j) is within the sensing range of the l^{th} mobile node.

Constraints (9) and (10) are used to map the individual coverage of mobile sensor nodes to the global coverage of the network area. It is worth noting that a grid cell could be covered more than once primarily because the traveling/movement range of mobile nodes is limited due to which multiple nodes may visit a grid cell across different time-steps to reach the uncovered cells. Thus, (9) ensures that a grid cell that has been covered in any time-step by any mobile node is considered to be covered throughout the rest of the algorithm. Constraint (10) sets $c_{i,j}$ to 0 if the grid cell (i, j) has not been covered by any mobile node during any time-step.

Overlapping coverage constraint: Due to the restriction on the one-step movement range of mobile nodes, it is possible for a grid cell to be covered more than once. We refer to this as overlapped/redundant coverage. To limit the overlapping coverage, constraint (11) is employed. This is similar to the overlapping coverage constraint in MILP-Static. This condition allows for limited overlap or redundancy in coverage (i.e., allows each cell to be covered at most $\hat{m}_o$ times) due to multiple mobile nodes covering the same cell across time-steps. This is essential to ensure that area coverage can be maximized while allowing some redundancy in coverage. Setting $\hat{m}_o = 1$ may result in the mobile node paths getting stuck at a cell before $K_{\max}$ time-steps at the cost of limiting the area coverage.

Constraint (12) defines the upper and lower limits for the continuous variables $c_{i,j}$ and $c_{i,j}^{l,k}$. This ensures that the value of these variables does not exceed 1, even if the corresponding grid points are covered multiple times.

C. Movement Minimization

An alternative strategy for planning the paths of the mobile nodes is to minimize the total number of movements by the mobile nodes for attaining a desired coverage ratio (cr). This can be formulated as the following optimization problem and is referred to as MILP-Mov.

$$\min \sum_{k=1}^{K_{\max}} \sum_{\substack{l=1 \\ (i,j)\in\overline{\mathcal{C}}_s}}^{N_m} x_{i,j}^{l,k} \tag{13}$$

$$\text{s.t.} \sum_{(i,j)\in\overline{\mathcal{C}}_s} x_{i,j}^{l,k} \leq 1; l = 1,\ldots,N_m; k = 1,\ldots,K_{\max} \tag{14}$$

$$\sum_{(i,j)\in\mathcal{C}} c_{i,j} \geq cr \cdot |\mathcal{C}|$$

$$\text{Constraints (7), (8), (9), (10), (11), (12)} \tag{15}$$

Objective function: The mobility strategy assumed for the mobile nodes allows each mobile node to undertake only one movement per time-step. The total number of movements, therefore, can be determined by the number of cells occupied by all the mobile nodes along their paths. Other than communication, a major part of energy is used for movements of the mobile nodes. Hence, minimizing the number of movements can save a significant amount of energy. The cost function in (13) minimizes the total number of cells that are visited/occupied by the mobile nodes across all time-steps.

The constraints in MILP-Mov are the same as in MILP-Cov with two changes. (14) is the position constraint, which is similar to the position constraint (6) of MILP-Cov, except that MILP-Mov allows the possibility that after a certain number of time-steps $x_{i,j}^{l,k}$ for all the mobile nodes can be zero. This implies that all the mobile nodes would stop their movements as soon as the desired coverage ratio is achieved.

Coverage ratio constraint: The mobile nodes traverse through the uncovered cells to ensure 1-*degree* periodic coverage (or 1-*coverage*). However, the mobile nodes need to stop after some time-steps. The stopping criteria for MILP-Cov was determined by the value of $K_{\max}$, whereas in this formulation it given by (15). This constraint indicates that the total coverage achieved by the static and mobile nodes must satisfy or exceed the desired area coverage ratio. With $cr = 1$, this algorithm is aimed at achieving 100% area coverage.

TABLE II
AREA COVERAGE (MILP-COV) USING TWO DIFFERENT STATIC NODE PLACEMENT STRATEGIES: RANDOM AND MILP-STATIC

Network Area	N_m	N_s	Area Coverage (%)	
			Random	MILP-Static
8×8	1	3	63.18	85.93
	1	5	68.23	100
	2	3	95.93	100
	2	5	93.39	100
	3	3	100	100
	3	5	94.47	100
10×10	1	3	42.12	60
	1	5	44.91	77
	2	3	75.11	88
	2	5	78.14	96
	3	3	93.24	99
	3	5	94.88	100

IV. SIMULATION RESULTS

In this section, we present detailed performance analysis of the proposed MILP-based node placement and path planning algorithms.

A. Simulation Setup

We consider network area of sizes ranging from 8×8 to 20×20 with grid cells of size 1×1. The number of static and mobile nodes are varied in the range 0–10 and 1–5, respectively. We assume sensing range $r_s = 1$, one-step traveling range $\rho_x = \rho_y = 2$, and overlapping cell coverage factor $\hat{m}_o = 3$ for MILP-Cov and MILP-Mov, and $\hat{s}_o = 1$ for MILP-Static. These parameter values are used unless specified otherwise. The algorithms were implemented in MATLAB R2020b and used 12.10.0 version of IBM ILOG CPLEX optimization software. The CPLEX parameter settings used include 'TimeLimit' of 18000 s, 'MIPGap' of 0, and branch and cut method as the 'MIP Strategy Search'.

B. Static Node Placement

We first compare the two static node placement strategies, random and MILP-Static, in terms of area coverage. We assume that after the static node placement, the paths of the mobile nodes are planned using MILP-Cov. Table II compares the network area coverage achieved using different numbers of mobile and static sensor nodes for two different network areas. In the random placement strategy, the static nodes are placed according to

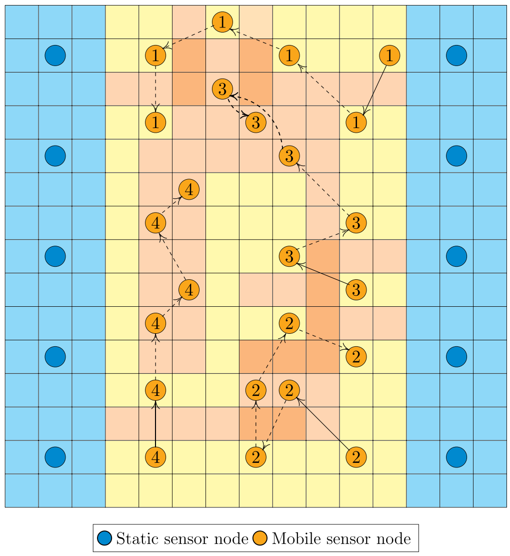


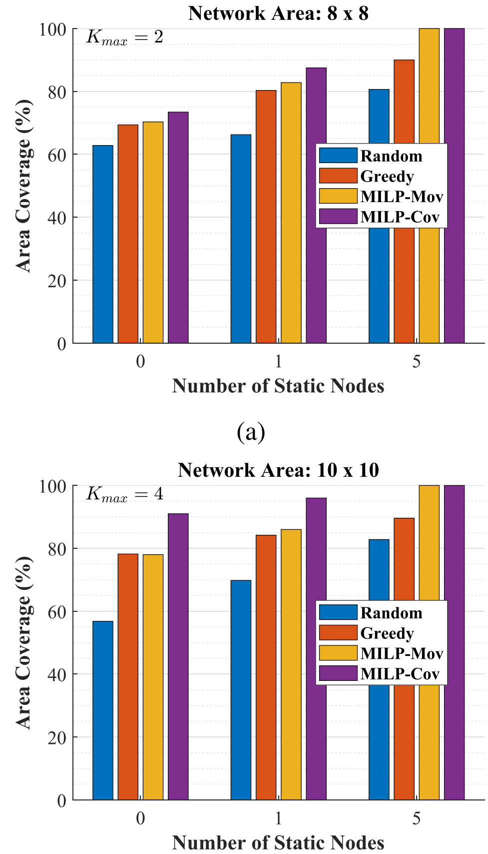
Fig. 3. Coverage path of four mobile nodes in a 15×15 network area with ten static sensor nodes deployed using MILP-Static, $r_s = 1$, $\rho_x = \rho_y = 2$, $\hat{s}_o = 1$, $\hat{m}_o = 3$, and $K_{\max} = 6$.

a uniform distribution. The area coverage values reported are obtained with $K_{\max} = 4$ and by averaging over five simulation runs. The results indicate that the area coverage performance is significantly better in the case where the static nodes are placed using the MILP-Static strategy, as compared to random placement. Based on these results, for the rest of the simulations of MILP-Cov and MILP-Mov, the MILP-Static algorithm will be used for the placement of static nodes. Fig. 3 shows the static node placement and paths of the mobile nodes obtained using MILP-Static and MILP-Cov, respectively. The arrows indicate the paths of the mobile nodes across time-steps and overlapping coverage is shown using increasing intensity of shades.

C. MILP-Cov and MILP-Mov Performance

In Fig. 4, we compare the performance of MILP-Cov, MILP-Mov, random movement, and greedy approaches in terms of the area coverage, with three mobile nodes. In the greedy approach, each mobile node considers the cells within its one-step traveling range and moves to the cell location that would cover the maximum number of cells which are yet to be covered. On the other hand, in the random movement approach, each mobile node moves to a cell that is selected randomly from all the potential cells within its one-step traveling range. It is seen that MILP-Cov consistently outperforms random and greedy approaches with the greedy approach performing better than the random movement approach. With $N_s = 0$ or 1, the area coverage using MILP-Mov and greedy approaches is similar since with this level of coverage through static node, the overall coverage search space does not change significantly. Table III considers the effect of sensing and traveling range on the area coverage. It is seen that increasing the traveling range results in a marginal improvement in the coverage performance of MILP-Cov. In contrast, a larger sensing range is seen to significantly improve the coverage performance.

In Fig. 5, we consider the performance of the MILP-Mov strategy in terms of the number of movements needed to achieve



(a)

(b)

Fig. 4. Area coverage using three ($N_m = 3$) mobile nodes with different number of static nodes.

TABLE III
EFFECT OF SENSING AND TRAVELING RANGE ON AREA COVERAGE USING MILP-COV IN A 20×20 NETWORK AREA WITH $N_s = 5$, $N_m = 3$, AND $K_{\text{MAX}} = 5$

r_s	$\rho_x = \rho_y$	Area Coverage (%)
1	1	33.00
	2	42.00
	3	45.00
	4	45.00
	5	45.00
	5 ($K_{\max} = 10$)	78.50
2	1 ($K_{\max} = 3$)	63.50
	2	91.25
	3	99.50
	4	100.00
	5	100.00

full area coverage ($cr = 1$) for two different network areas. The results show that, for a given network area, as the number of static and mobile nodes increases, the required number of movements decreases. In Fig. 6, we compare performance of the MILP-Cov, MILP-Mov algorithms, and the path planning technique presented in [17], in terms of the area coverage as a function of the number of mobile node movements. We refer to the method presented in [17] as the Vecchio method, based on the first author's last name. For a fair comparison, the initial locations of the static and mobile nodes are kept the same for

TABLE IV
COMPARISON OF THE NUMBER OF CONSTRAINTS AND VARIABLES

Method	No. of Binary Variables	No. of Continuous Variables	No. of Constraints
MILP-Cov	$N_m K_{\max}\|\mathcal{C}\|$	$(1 + N_m K_{\max})\|\mathcal{C}\|$	$N_m K_{\max}(3\|\mathcal{C}\| + 1) + \|\mathcal{C}\|(2 - N_m)$
MILP-Mov	$N_m K_{\max}\|\mathcal{C}\|$	$(1 + N_m K_{\max})\|\mathcal{C}\|$	$N_m K_{\max}(3\|\mathcal{C}\| + 1) + \|\mathcal{C}\|(2 - N_m) + 1$
MILP-Static	$N_s\|\mathcal{C}\|$	$N_s\|\mathcal{C}\|$	$N_s + (N_s + 1)\|\mathcal{C}\|$

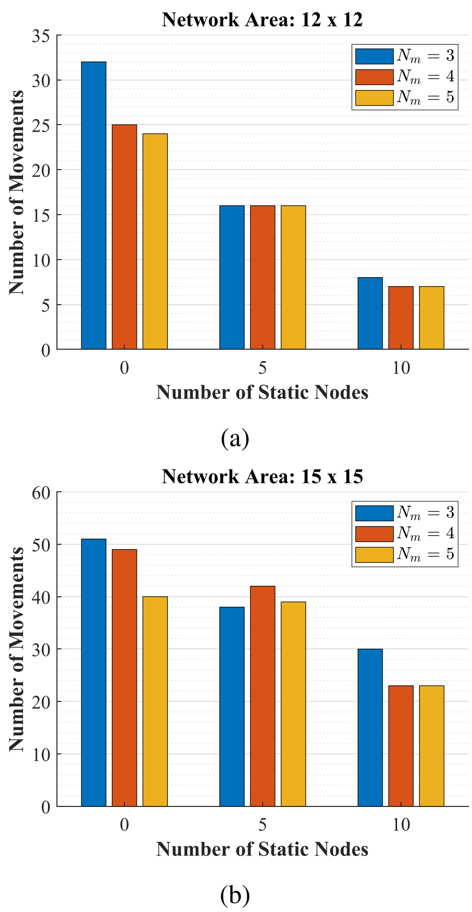


Fig. 5. Number of movements required by mobile nodes for full area coverage ($cr = 1$) using MILP-Mov with different number of static and mobile nodes.

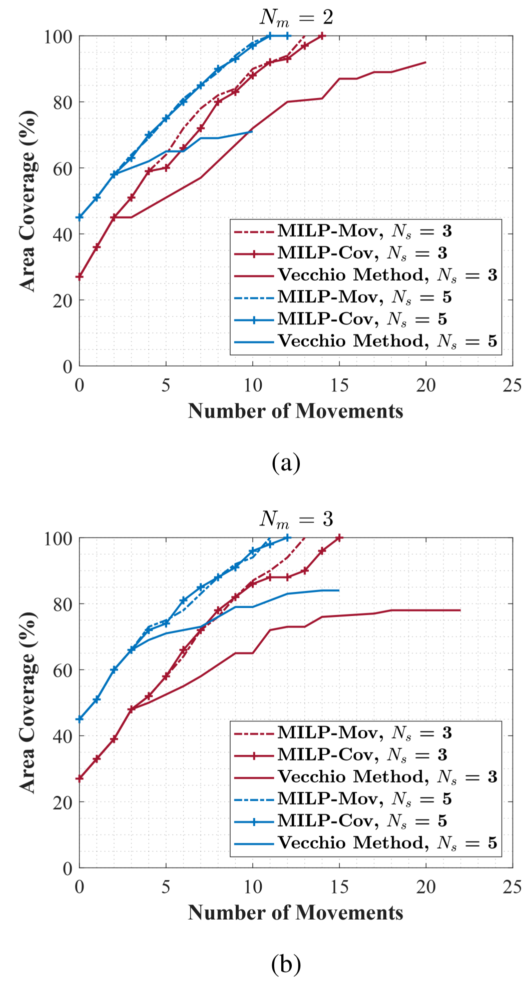


Fig. 6. Area coverage as a function of the number of movements by the mobile nodes (Network area $= 10 \times 10$).

all the three algorithms. The initial locations of static and mobile nodes are obtained based on MILP-Static and MILP-Mov algorithms, respectively. For implementing the Vecchio method, we use $r_s = 1.5$, $r_c = 2r_s$, $\mu = 2$, $\rho = 2$, $\phi = \pi/6$, $n = 10$, and these parameters have the same definitions as in [17]. The values of weight parameters associated with the cost function and other parameters are as given in [17]. From Fig. 6, it is seen that the proposed MILP-Cov and MILP-Mov algorithms achieve full area coverage, and outperform the Vecchio method [17] by requiring much fewer mobile node movements for achieving a given level of area coverage. It is observed that the Vecchio method does not achieve full area coverage even with a very large number of movements and the coverage saturates at some point. In addition, in the Vecchio method, the mobile nodes stop their movements before achieving full coverage as they are unable to locate the uncovered areas and wait endlessly while looking for the next location using the zoom algorithm [19], [20]. Although, the three methods show improved coverage performance with increase in the number of static and mobile nodes.

D. Computational Complexity Analysis

We analyze the complexity of the proposed methods in terms of the number of continuous variables, number of integer variables, and the number of constraints. These are listed in Table IV and it is seen that these complexity measures increase linearly with the network area and the number of nodes in the network. In Table V, we list the number of movements along with the computational time required for full coverage by MILP-Cov and MILP-Mov methods for different network areas and with different number of static and mobile nodes. The static node placement is performed using the proposed MILP-Static strategy. The simulations are performed on an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz-1.99 GHz, 16 GB RAM, and running Microsoft Windows 10 Pro. The number of constraints in MILP-Cov and MILP-Mov only differ by one, however MILP-Cov has a consistently lower CPU time

TABLE V
COMPARISON OF THE NUMBER OF MOVEMENTS AND CPU TIMES REQUIRED FOR FULL COVERAGE BY MILP-COV AND MILP-MOV

Network Area	N_m	N_s	Movements		CPU Time (s)	
			MILP-Cov	MILP-Mov	MILP-Cov	MILP-Mov
10×10	3	0	17	16	7.2	1573.6
		5	12	11	1.3	6.4
		10	8	6	1.5	2.3
12×12	3	0	32	32	454.2	18000
		5	18	16	16.9	18000
		10	9	8	3.4	117.2
10×10	4	0	16	16	3.6	1536.8
		5	12	11	1.2	31.9
		10	7	6	1.4	2.9
12×12	4	0	28	25	137.6	18000
		5	16	16	9.0	18000
		10	10	7	3.6	164.8
10×10	5	0	16	16	5.6	3326.7
		5	11	11	1.3	188.4
		10	8	6	1.4	3.2
12×12	5	0	25	24	118.3	18000
		5	18	16	10.4	18000
		10	10	7	3.4	15.6

than MILP-Mov. This is because, in MILP-Mov, the objective function involves binary variables and the complexity of ILP increases exponentially with integer/binary variables.

V. CONCLUSION

We have proposed three MILP-based formulations, MILP-Static, MILP-Cov, and MILP-Mov, for efficient placement of static nodes and path planning of mobile nodes in order to maximize the network area coverage and minimize the number of mobile node movements to achieve a desired area coverage. The static node placement strategy addresses the issues that can arise due to random deployment of static nodes and the path planning strategies allow for an explicit limit on overlapping/redundant coverage. The proposed path planning methods achieve improved area coverage and improve the sweep coverage time by minimizing the number of mobile node movements, which in turn can improve the network lifetime.

REFERENCES

[1] A. V. Savkin and H. Huang, "A method for optimized deployment of a network of surveillance aerial drones," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4474–4477, Dec. 2019.
[2] T. P. Lambrou and C. G. Panayiotou, "Collaborative path planning for event search and exploration in mixed sensor networks," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1424–1437, 2013.
[3] B. Liu, P. Braß, O. Dousse, P. Nain, and D. Towsley, "Mobility improve coverage of sensor networks," in *Proc. Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2005, pp. 300–308.
[4] W. Liang, J. Luo, and X. Xu, "Prolonging network lifetime via a controlled mobile sink in wireless sensor networks," in *Proc. IEEE Glob. Telecommun. Conf.*, 2010, pp. 1–6.
[5] P. Zhong and F. Ruan, "An energy efficient multiple mobile sinks based routing algorithm for wireless sensor networks," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 323, 2018, Art. no. 012029.
[6] X. Gao, J. Fan, F. Wu, and G. Chen, "Cooperative sweep coverage problem with mobile sensors," *IEEE Trans. Mobile Comput.*, vol. 21, no. 2, pp. 480–494, Feb. 2022.
[7] X. Gao, X. Zhu, Y. Feng, F. Wu, and G. Chen, "Data ferry trajectory planning for sweep coverage problem with multiple mobile sensors," in *Proc. IEEE 13th Annu. Int. Conf. Sens. Commun. Netw.*, 2016, pp. 1–9.
[8] A. Mohsen, W. Aljoby, K. Alenezi, and A. Alenezi, "A robust harmony search algorithm based Markov model for node deployment in hybrid wireless sensor networks," *Int. J. GEOMATE*, vol. 11, no. 5, pp. 2747–2754, 2016.
[9] L. Zhu, C. Fan, H. Wu, and Z. Wen, "Coverage optimization algorithm of wireless sensor network based on mobile nodes," *Int. J. Online Biomed. Eng.*, vol. 12, no. 08, pp. 45–50, 2016.
[10] O. Banimelhem, M. Mowafi, and W. Aljoby, "Genetic algorithm based node deployment in hybrid wireless sensor networks," *Commun. Netw.*, vol. 5, pp. 273–279, 2013.
[11] L. Kong, K. Ma, B. Qiao, and X. Guo, "Adaptive relay chain routing with load balancing and high energy efficiency," *IEEE Sens. J.*, vol. 16, no. 14, pp. 5826–5836, Jul. 2016.
[12] K. Lee, Y.-H. Kim, H.-J. Kim, and S. Han, "A myopic mobile sink migration strategy for maximizing lifetime of wireless sensor networks," *Wireless Netw.*, vol. 20, no. 2, pp. 303–318, 2014.
[13] R. Elhabyan, W. Shi, and M. St-Hilaire, "Coverage protocols for wireless sensor networks: Review and future directions," *IEEE J. Commun. Netw.*, vol. 21, no. 1, pp. 45–60, Feb. 2019.
[14] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, "A survey on mobility in wireless sensor networks," *Ad Hoc Netw.*, vol. 125, 2022, Art. no. 102726.
[15] X. Gao, J. Fan, F. Wu, and G. Chen, "Approximation algorithms for sweep coverage problem with multiple mobile sensors," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 990–1003, Apr. 2018.
[16] G. Wang, G. Cao, P. Berman, and T. F. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Trans. Mobile Comput.*, vol. 6, no. 5, pp. 563–576, May 2007.
[17] M. Vecchio and R. López-Valcarce, "Improving area coverage of wireless sensor networks via controllable mobile nodes: A greedy approach," *J. Netw. Comput. Appl.*, vol. 48, pp. 1–13, 2015.
[18] T. Lambrou, C. Panayiotou, S. Felici-Castell, and B. Beferull-Lozano, "Exploiting mobility for efficient coverage in sparse wireless sensor networks," *Wireless Pers. Commun.*, vol. 54, pp. 187–201, 2010.
[19] T. P. Lambrou and C. G. Panayiotou, "Collaborative event detection using mobile and stationary nodes in sensor networks," in *Proc. Int. Conf. Collaborative Comput. Netw. Appl. Worksharing*, 2007, pp. 106–115.
[20] T. P. Lambrou and C. G. Panayiotou, "Collaborative area monitoring using wireless sensor networks with stationary and mobile nodes," *J. Adv. Signal Process.*, vol. 2009, pp. 1–16, 2009.
[21] C. Zygowski and A. Jaekel, "Optimal path planning strategies for monitoring coverage holes in wireless sensor networks," *Ad Hoc Netw.*, vol. 96, 2020, Art. no. 101990.
[22] C. Cav and A. Altin-Kayhan, "Coverage hole optimization with a mobile sensor in wireless sensor networks for smart grid," *Ad Hoc Netw.*, vol. 140, 2023, Art. no. 103039.
[23] M. Watfa and S. Commuri, "Boundary coverage and coverage boundary problems in wireless sensor networks," *Int. J. Sens. Netw.*, vol. 2, pp. 273–283, 2007.