Sensor Networks for Structural Health Monitoring of Critical Infrastructures Using Identifying Codes

Kaustav Basu*, Sanjana Dey†, Subhas Nandy†, Arunabha Sen*
*NetXT Lab, SCIDSE, Arizona State University
{kaustav.basu, asen}@asu.edu
†Indian Statistical Institute, Kolkata
{info4.sanjana, subhas.c.nandy}@gmail.com

***Abstract*—The Structural Health Monitoring (SHM) problem for critical infrastructures using wireless sensor networks (WSN), has received considerable attention in the research community in recent years. Sensors placed in these infrastructures have two functions, *sensing/coverage* and *communication*. The thrust of this paper is on the coverage aspects of sensor networks. In the Point Coverage model, only a specified set of points in the deployment area have to be sensed. The goal of placement optimization is to find the smallest set of locations to deploy sensors, so that all the points of interest can be sensed. This problem often is solved by formulating it as a Set Cover problem. However, the Set Cover approach has a serious limitation on the accurate identification of the location where abnormality is sensed. In this paper, we present a technique to overcome this limitation by utilizing Identifying Code. We study two different scenarios, where the sensors and points of interest are located in one and two-dimensional spaces respectively. We provide a polynomial time optimal algorithm for the one-dimensional case and an Integer Linear Programming (ILP) based optimal solution for the two-dimensional case. We evaluate the efficacy of the ILP solution with varying network size (45 to 64655 nodes). The ILP produced an optimal solution for the largest instance with 64655 nodes and 155339 edges in only 180.45 seconds.**

***Index Terms*—Structural Health Monitoring, Identifying Code, Discriminating Code, Interval Bigraph, Polynomial Algorithm**

I. Introduction

The Structural Health Monitoring (SHM) problem for critical infrastructures, such as bridges, buildings, electric power equipments, using wireless sensor networks, has received considerable attention in the research community in recent years [1]. Sensors placed in the deployment area have two functions: (i) to sense a target function such as temperature, pressure, vibration, etc., and (ii) to transmit the sensed data either directly or through multiple other sensor nodes (which serves as relays) to the control station, for the analysis of the sensed data. While the first function relates to coverage of the sensing region, the second function relates to the connectivity aspects of the network formed by the sensors. The thrust of this paper is on the coverage aspects of the sensors distributed over a geographic region. The sensors used for monitoring critical infrastructures may be *accelerometers, strain sensors, corrosion sensors, linear voltage differential transducers* and *optical fiber transducers* [1]. Each of these sensors have a specific *sensing range* associated with it. A sensor can only sense any abnormality if it happens within its sensing range, i.e., each sensor has a *coverage area* associated it. The coverage aspect of sensor networks alone has been studied extensively [2], [3]. The survey paper on Coverage Problems in Sensor Networks [2], references close to two hundred papers. Accordingly, a multitude of sensor coverage models, such as (i) Boolean Sector Coverage Model, (ii) Boolean Disc Coverage Model, (iii) Attenuated Disc Coverage Model, (iv) Truncated Attenuated Disc Models, (v) Estimation Coverage Models, etc. have been studied by various research groups [2].


Fig. 1. Structural Monitoring of Bridges

Cardei and Wu in [3], classified coverage problems into three broad classes, (i) Point Coverage, (ii) Area Coverage, and (iii) Barrier Coverage. While in the area coverage problem, an entire area (in two or three dimensional space) has to be sensed (monitored), in the point coverage problem, only a *specified set of points* (points of interest) in two or three dimensional space, has to be monitored. Moreover, oftentimes there are restrictions on the locations (in two or three dimensional space), where the sensors can be deployed. These are the potential locations where the sensors can be deployed, and these locations can be viewed as another set of specified points (in a two or three dimensional space). For example, accelerometers cannot be deployed in any arbitrary location, due to various constraints such as, cost of deployment, access difficulty, property rights, etc.

One of the most frequently studied problems in this context is the Sensor Placement Optimization [4], whose goal is to find the smallest set of locations (among the potential locations for sensor deployment), so that all the points of interest can be sensed. In other words, every point of interest should be under the *coverage* area of at least one deployed sensor. If a boolean disc coverage model [2] is used for sensor coverage, the Sensor Placement Optimization problem can be formulated as a Set Cover problem [4] and a number of studies using this model

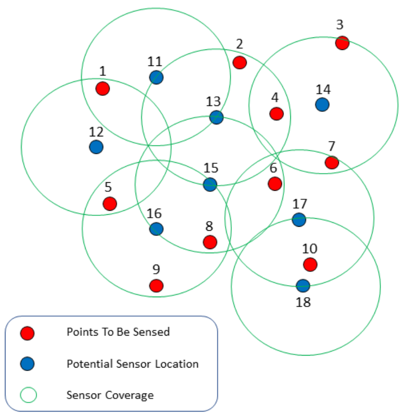

Fig. 2. Potential Sensors and Sensing Locations

TABLE I
POINTS COVERED BY EACH SENSOR

Sensor Location	Points Sensed	Sensor Location	Points Sensed
11	1	15	6, 8
12	1, 5	16	5, 8, 9
13	2, 4	17	6, 7, 10
14	3, 4, 7	18	10

are available in the literature [4].

Although a number of studies on sensor placement optimization problem follow the set cover formulation to find a solution, it has a serious limitation on the *accurate* identification of the location, where some abnormality is detected by one or more of the deployed sensors. We illustrate this point with the help of an example. In Fig. 2, the ten red points (numbered from 1-10) indicate the points to be sensed (monitored), the eight blue points (numbered from 11-18) indicate the potential locations where the sensors can be deployed and the green circles (centered on each blue point) indicate the coverage area of a sensor deployed at that blue point. In this example, it can be verified that if sensors are deployed in locations 11, 13, 14, 16 and 17, all points from 1 to 10 will be within the sensing range of at least one sensor. Specifically, the points (1-10) covered (sensed) by the sensors $11, 13, 14, 16, 17$ are shown in Table. I. In Table. II, we present the sensors that are actually sensing the points 1-10, using the Set Cover approach.

The serious limitation of the set cover based approach to optimal sensor placement problem is that, it may fail to uniquely identify the point where an abnormality is detected by the sensor. We elaborate this point with the results shown in Tables. I and II. In this example, sensors were deployed at locations $11, 13, 14, 16, 17$ and this deployment ensured that all points to be sensed were within the coverage area of at least one sensor. Suppose the control center has five indicator lamps A, B, C, D, E corresponding to five sensors located at $11, 13, 14, 16, 17$. If the sensor does not sense an abnormality at the point it is sensing, then the corresponding lamp is lit green. If a sensor senses an abnormality at a point, then the corresponding lamp turns red. From Table. II, it can be seen that points 6 and 10 are sensed by sensor 17 only, and points

TABLE II
SENSORS COVERING EACH POINT

Points Sensed	Sensor Location	Points Sensed	Sensor Location
1	11	6	17*
2	13	7	14, 17
3	14	8	16**
4	13, 14	9	16**
5	16**	10	17*

TABLE III
SENSORS COVERING EACH POINT

Points Sensed	Sensor Location	Points Sensed	Sensor Location
1	12	6	15, 17
2	13	7	14,17
3	14	8	15, 16
4	13, 14	9	16
5	12, 16	10	17

5, 8 and 9 are sensed by 16 only. The implication of this is that if lamp E (corresponding to sensor 17) turns red, then it will not be possible to ascertain if the abnormality was detected at point 6 or 10. Similarly, if lamp D (corresponding to sensor 16) turns red, then it will not be possible to ascertain if the abnormality was detected at point 5 or 8 or 9.

This limitation of failure to uniquely identify the point where abnormality is detected by the sensor, can be overcome by deployment of additional sensors. In this example, instead of deploying sensors at locations $11, 13, 14, 16, 17$, if they were deployed at locations $12, 13, 14, 15, 16, 17$, then each point would have been sensed in the way as shown in Table. III. It may be noticed that deployment of six sensors, instead of five, avoids the problem of failure of unique identification of points where abnormality is detected.

The mathematical foundation of computing the least number of sensors needed for unique identification of points where abnormality is detected, is provided by the concept of Identifying and Discriminating Codes [13]. Since its introduction, these codes have been studied fairly extensively for various classes of graphs. We show that the problem of computing the least number of sensors needed for unique identification of points, where abnormality is detected, for points in one-dimensional and two-dimensional space is equivalent to computation of *Minimum Discriminating Code Set (MDCS)* of *Unit Interval Bigraph* and *Unit Disc Bigraph* respectively. A major contribution of this paper is the development of a polynomial time algorithm for the MDCS problem for Unit Interval Bigraphs. It may be noted that when the intervals associated with the nodes of the graph are of arbitrary size, then the MDCS problem is NP-Complete [14]. Computational complexity of related problems, such as Identifying Code for Interval Graphs is NP-Complete, whereas it is still open for Interval Graphs with unit length intervals [15].

II. RELATED WORK

As noted earlier, sensor placement problems for monitoring critical infrastructures have been studied extensively in the last few years [5]–[12]. A particular example of health monitoring in civil infrastructures using wireless sensor networks can be seen in [9], where the authors deployed sensors to monitor the Golden Gate Bridge. In this application, both the points to be monitored and the locations where the sensors can be deployed, lie on a single line, as shown in Fig. 5. Our study on accurate identification of the location where abnormality is sensed, in which sensors and points of interest are located on one line, is motivated by the study in [9]. WSN based SHM system is also deployed at the many bridges in China including the ZhengDian bridge [1]. Aside from bridges, SHM systems have also been deployed for monitoring sports stadium, buildings and wind turbines [1].

Karpovsky *et. al.* introduced the concept of Identifying Codes in [13] and provided results for Identifying Codes for graphs with specific topologies, such as binary cubes and trees. Using Identifying Codes, Laifenfeld *et. al.* studied joint monitoring and routing in wireless sensor networks, in [16]. Ray *et. al.* studied location detection problem in emergency sensor networks, using Identifying Codes [17]. In this paper, they also introduced the concept of robust Identifying Codes to deal with faults in sensor networks. They presented an algorithm for generating *irreducible* Identifying Codes in polynomial time. It may be noted that irreducible Identifying Code is only a *minimal* Identifying Code and may not be the *minimum* (or optimal) Identifying Code. In contrast, we present an algorithm for construction of optimal Identifying Code for the problem scenario, where the points of interest and potential locations for sensor deployment, lies on a single line. Sen *et. al.* studied monitoring terrorist networks using Identifying Codes [18].

Algorithms and complexity of computation of Identifying Codes for restricted class of graphs such as, Interval and Permutation graphs, were studied in [19]. An approximation algorithm for the computation of minimum Identifying Code for Interval graphs with a performance bound of six, was presented in [15]. A special case, where only a subset of nodes needs a unique code, can be modeled with a bipartite graph, and this version of Identifying Codes is called "Discriminating Codes" and was studied in [20]. This special case is relevant for our study as, our problem formulation requires us to find the unique signatures of all nodes of one side of bipartition of a bipartite graph, by selecting only a subset of the nodes in the other side of bipartition. This formulation corresponds directly to "Discriminating Codes". However, the bipartite graphs that appear in our formulation is not just any bipartite graph, but subsets of bipartite graphs known as Interval Bigraphs and Unit Disc Bigraphs, corresponding to our study of points being in one or two dimensional space respectively. Muller in [21] presented a polynomial time algorithm for recognition of Interval Bigraphs.

III. Overview of Graphs and Codes

The notion of *Identifying Codes* was proposed in [13] and has turned out to be a useful concept for optimizing sensor deployment in multiple domains. In this paper, we use Identifying Code of the *simplest form*, and define it as follows. *A vertex set V' of a graph $G = (V, E)$ is defined as the Identifying Code Set (ICS) for the vertex set V, if for all $v \in V$, $N^+[v] \cap V'$ is unique where, $N^+[v] = v \cup N(v)$ and $N(v)$ represents the set of nodes adjacent to v in $G = (V, E)$.* The *Minimum Identifying Code Set* (MICS) problem is to find the Identifying Code Set of *smallest cardinality*. The vertices of the set V' may be viewed as *alphabets* of the code, and the *string* made up by concatenation of the alphabets of $N[v]$, may be viewed as the unique "code" for the node v. For instance, consider the graph $G = (V, E)$ shown in Fig. 3. In this graph $V' = \{v_1, v_2, v_3, v_4\}$ is an ICS, as it can be seen from Table IV that $N^+[v] \cap V'$ is *unique* for all $v_i \in V$. From the table, it can be seen that the code for node v_1 is v_1, the code for v_5 is v_1, v_2, the code for v_{10} is v_3, v_4, etc. It may be noted that some graphs may not have an Identifying Code Set. Two nodes $u, v \in V$ are said to be "twins" if $N^+[v] = N^+[u]$. It may be noted that Identifying Code for a graph $G = (V, E)$ does not exist if the graph has "twins".

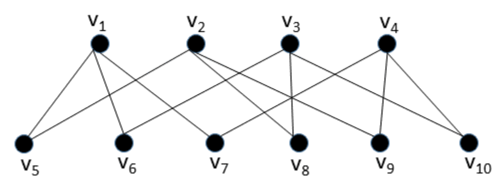

Fig. 3. Graph with Identifying Code Set $\{v_1, v_2, v_3, v_4\}$

TABLE IV
$N^+[v] \cap V'$ RESULTS FOR ALL $v \in V$ FOR THE GRAPH IN FIG. 3

$N^+[v_1] \cap V' = \{v_1\}$	$N^+[v_2] \cap V' = \{v_2\}$
$N^+[v_3] \cap V' = \{v_3\}$	$N^+[v_4] \cap V' = \{v_4\}$
$N^+[v_5] \cap V' = \{v_1, v_2\}$	$N^+[v_6] \cap V' = \{v_1, v_3\}$
$N^+[v_7] \cap V' = \{v_1, v_4\}$	$N^+[v_8] \cap V' = \{v_2, v_3\}$
$N^+[v_9] \cap V' = \{v_2, v_4\}$	$N^+[v_{10}] \cap V' = \{v_3, v_4\}$

`Graph Coloring with Seepage (GCS) Problem`: The MICS computation problem can be viewed as a novel variation of the classical Graph Coloring problem. We will refer to this version as the *Graph Coloring with Seepage (GCS)* problem. In the classical graph coloring problem, when a color is *assigned* (or injected) to a node, only that node is colored. The goal of the classical graph coloring problem is to use as few distinct colors as possible such that (i) every node receives a color, and (ii) no two adjacent nodes of the graph have the same color. In the GCS problem, when a color is assigned (or injected) to a node, not only does that node receive the color, but also the color *seeps* into all the adjoining nodes. For example, if a node v_i is adjacent to two other nodes v_j and v_k in the graph, then if the color red is injected to v_j, not only v_j will become red, but also v_i will become red as it is adjacent to v_j. Now if the color blue is injected to v_k, not only v_k will become blue, but also the color blue will seep in to v_i as it is adjacent to v_k. Since v_i was already colored red (due to seepage from v_j), after color seepage from v_k, it's color will be a *combination of red and blue (purple)*. At this point all three nodes v_j, v_k and v_i have "distinct" colors red, blue, and purple, respectively. *The color assigned to a node may be due to, (i) only injection to that*

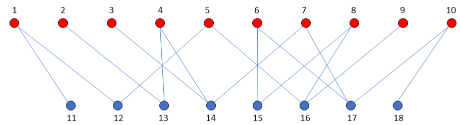

Fig. 4. Graph corresponding to the problem instance shown in Fig. 2

node, (ii) only seepage from other adjoining nodes and (iii) a combination of injection and seepage. The colors injected at the nodes will be referred to as *atomic* colors. The colors formed by the combination of two or more atomic colors are referred to as *composite* colors. The colors injected at the nodes (atomic colors) are all *unique*. The goal of the GCS problem is to inject colors to as few nodes as possible, such that (i) every node receives a color, and (ii) no two nodes of the graph have the same color.

Suppose that the node set V' is an ICS of a graph $G = (V, E)$ and $|V'| = p$. In this case, if p distinct colors are injected to V' (one distinct atomic color to one node of V'), then as per the definition of ICS, for all $v \in V$, if $N^+[v] \cap V'$ is unique, all nodes of $G = (V, E)$ will have a unique color (either atomic or composite). Thus computation of MICS is equivalent to solving the GCS problem.

A variation of Identifying Code, when restricted to Bipartite graphs, is known as *Discriminating Code* [20], and is defined as follows: Let $G = (V_1 \cup V_2, E)$ be an undirected bipartite graph and let $N(v)$, denote the neighborhood of v, for any $v \in V_1 \cup V_2$. A subset $V_2' \subseteq V_2$ is called the Discriminating Code of G, if $\forall v \in V_1, N(v) \cap V_2'$ is unique.

Minimum Discriminating Code Set (MDCS) Problem: Find the smallest subset $V_2' \subseteq V_2$, such that injection of colors at these nodes, ensures that each node $v \in V_1$, receives a unique color through seepage.

In the following, we define a set of graphs that are relevant for this study.

Definition III.1. *Interval Bigraphs: An interval bigraph is an undirected bipartite graph $G = (V_1 \cup V_2, E)$, whose edge set is the intersection of the edge set of an interval graph with vertex set $V_1 \cup V_2$, and the edge set of a complete bipartite graph with bipartition $V_1 \cup V_2$. A bipartite interval representation of an interval bigraph is given by a bipartitioned set of intervals for its vertices, such that vertices are adjacent if and only if the corresponding intervals intersect, and belong to opposite sides of the bipartition [21].*

Definition III.2. *Unit Disc Bigraphs: A unit disc bigraph is an undirected bipartite graph, whose edge set is the intersection of the edge sets of a unit disc graph and the edge set of a complete bipartite graph on the same vertex set. A bipartite unit disc representation of a unit disc bigraph is given by a bipartitioned set of circles for its vertices, such that vertices are adjacent if and only if the corresponding circles intersect and belong to opposite sides of the bipartition.*

IV. PROBLEM FORMULATION

In Fig. 2, the ten red points (numbered from 1-10) indicate the points to be sensed (monitored), the eight blue points (numbered from 11-18) indicate the potential locations where the sensors can be deployed and the green circles (centered on each blue point) indicate the coverage area of a sensor deployed at that blue point. From these set of red and blue points, we construct a graph using the following construction rules: (i) Corresponding to each red point, we have a red node, (ii) Corresponding to each blue point, we have a blue node and (iii) There is an edge between a blue node and a red node, if and only if the corresponding red point is within the green circle, centered at the corresponding blue point. The graph constructed from the problem instance in Fig. 2, is shown in Fig. 4. Clearly, the graph constructed by following the above construction rules will result in a Bipartitie graph. However, a pertinent question in this regard is whether any Bipartite graph can be constructed by following the rules or the constructed graphs constitute only a *subset* of all Bipartite graphs. In the following theorem, we prove that the constructed graphs constitute only a *subset* of all Bipartite graphs.

Lemma 1. *The number of distinct regions created by intersection of n circles on a plane is at most $n^2 - n + 2$.*

Proof. The lemma is proven by induction. If $n = 1$ there are two regions (one inside and the other outside the circle), and the inductive hypothesis holds as $n^2 - n + 2 = 1 - 1 + 2 = 2$. Assume that the hypothesis is true as long as $n \leq m$. If a new $(m + 1)$-th circle is introduced, it can cross the old ones in at most 2m points. Each segment can cut an existing region in two parts, adding 2m regions. Thus the maximum number of regions after introduction of the for $(m + 1)$-th circle is $m^2 - m + 2 + 2m = (m + 1)^2 - (m + 1) + 2$. □

Theorem 2. *The graphs constructed by following the above construction rules only create a subset of all Bipartite graphs.*

Proof. The theorem can be proven by showing that there exists at least one Bipartite graph that cannot be constructed by following the construction rules.

Consider a Bipartite graph $G = (V_1 \cup V_2, E)$, where $V_1 \cup V_2$ constitutes the bipartition and cardinalities of V_1 and V_2 are n and $2^n - 1$ respectively. Each node in V_2 corresponds to a *non-empty* subset of the nodes in V_1 and is connected by an edge $e \in E$ to those nodes.

Claim: Such a Bipartite graph cannot be constructed using the graph construction rules given above. The reason such a graph cannot be constructed is the following. The node sets V_1 and V_2 corresponding to two sets of points on a two dimensional plane. Suppose that we refer to the set of points corresponding to the nodes in V_1 as red points and the set of points corresponding to the nodes in V_2 as blue points. Suppose that we draw a unit radius circle with each red point as the center. Since each node $v \in V_2$ corresponds to a distinct non-empty subset of nodes of V_1 and is connected to this subset of nodes, in order to satisfy the graph construction rules, the blue point corresponding to $v \in V_2$ must be located in a distinct region created by intersection of n circles corresponding to n red points. However, from Lemma 1 we know that the number of distinct regions created by intersection of n circles on a plane is at most $n^2 - n + 2$. For existence of the graph

$G = (V_1 \cup V_2, E)$, $2^n - 1$ distinct regions are needed as the locations of the $2^n - 1$ blue points. As $2^n - 1 > n^2 - n + 2$ for $n \geq 4$, the graph $G = (V_1 \cup V_2, E)$, such locations cannot be found when $n \geq 4$ and hence the graph cannot be constructed using the graph construction rules given above. □

Theorem 3. *The graph constructed from the two sets of points (red and blue) on a two dimensional plane using the construction rules given above will be an Unit Disc Bigraph.*

Proof. Clearly, the graph constructed following the construction rules will be a bipartite graph $G = (V_1 \cup V_2, E)$, where V_1 is the set of nodes corresponding to the set of red points and V_2 is the set of nodes corresponding to the set of blue points. Suppose that G_{udg} is a unit disc graph corresponding to the set of red and blue points, and G_{cbp} is a complete bipartite graph with bipartition V_1 and V_2. It can be easily verified that $G = (V_1 \cup V_2, E)$ is the *intersection* of G_{udg} and G_{cbp}. If $G_{udg} = (V_1 \cup V_2, E_{udg})$ and $G_{cbp} = (V_1 \cup V_2, E_{cbp})$, then $G = (V_1 \cup V_2, E_{udg} \cap E_{cbp})$. □

Theorem 4. *The graph constructed from the two sets of points (red and blue) on an one dimensional plane using the construction rules given above will be an Interval Bigraph.*

Proof. As in Lemma 3, the graph constructed following the construction rules will be a bipartite graph $G = (V_1 \cup V_2, E)$. Suppose that G_{ig} is an interval graph corresponding to the set of red and blue points and G_{cbp} is a complete bipartite graph with bipartition V_1 and V_2. It can be easily verified that $G = (V_1 \cup V_2, E)$ is the intersection G_{ig} and G_{cbp}. □

In this paper, we focus on two deployment scenarios. In the first case, both the points of interest and the potential sensor deployment locations are on a two dimensional space (i.e., a plane). In the second case, they are located on an one dimensional space (i.e., a line). The motivation for considering the scenario where they are located on an one dimensional space comes from the fact that, in a large number of structures (such as long spanning bridges), both the locations to be sensed and the locations where sensors can be deployed, lie on a line. In a study undertaken by a University of California Berkeley team, a wireless sensor network (WSN) was deployed for structural health monitoring (SHM) of the Golden Gate bridge, where the sensors were placed on a line as shown in Fig. 5. Similar efforts were undertaken in China as reported in [1].

It may be recalled that the main objective of this study is, to determine the *least* number of sensors and their locations, required for *unique identification* of the point of interest, where abnormality is detected. It may be noted that, in this study we focus our attention only to the scenario where abnormality (or failure) is restricted to only one point. In this paper, we do not consider multiple simultaneous failure.

It is clear from the discussion on the graph construction rules for the points deployed in two or one dimensional spaces that, the corresponding graphs will be a *Unit Disc Bigraph* and a *Interval Bigraph* respectively. It may be recalled that,

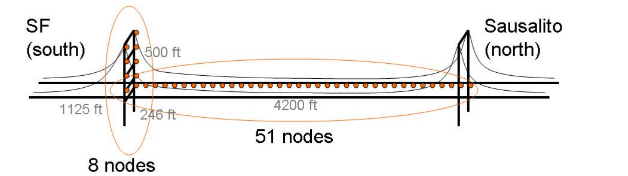

Fig. 5. Golden Gate Wireless Sensor Network [9]

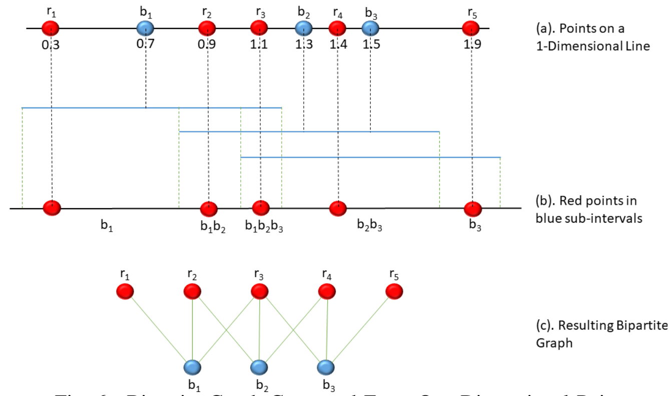

Fig. 6. Bipartite Graph Generated From One Dimensional Points

the MDCS Problem finds the smallest subset $V_2' \subseteq V_2$, such that injection of colors at these nodes, ensures that each node $v \in V_1$, receives a unique color through seepage. Thus, the objective of determining the *least* number of sensors required for *unique identification* of the point of failure can be realized, by the computation of MDCS for the Unit Disc Bigraph and Interval Bigraph, respectively.

V. Problem Solution

In this section, we first present a dynamic programming based optimal algorithm, for the computation of MDCS for points located in one dimensional space. We then present an Integer Linear Program for the computation of the MDCS for points located in two dimensional space.

A. One Dimensional Case

In this subsection, we provide a dynamic programming based algorithm to find the optimal solution for the problem in one dimension. Given a set of red points $R = \{r_1, ...r_n\}$ and a set of blue points $B = \{b_1, ..., b_m\}$ on a line, as shown in Fig. 6a, the dynamic programming algorithm finds the smallest subset $B' \subset B$, such that, injection of colors at these points, assigns a *unique* color (atomic or composite) to every red point in R through seepage. It may be recalled that the set of red points correspond to the points of interest and the set of blue points are potential locations for placement of sensors. The sensing range of a sensor is 1 unit, which implies that a blue point $b_i \in B$ can sense all the red points within the interval $b_i - 1$ to $b_i + 1$. The graph constructed from the set of points in Fig. 6a, following the graph construction rules mentioned earlier, is shown in Fig. 6c.

We use the notation R_j to be the $j-$th prefix of R, i.e., $R_j = \{r_1, ...r_j\}$. Similarly, B_i to be the $i-$th prefix of B, i.e., $B_i = \{b_1, ...b_i\}$. Since, the objective of this problem is to assign a *unique* color to all elements of R by injecting colors at the fewest number of elements of B, we refer to the R

points as the *problem space* and B points as the *solution space*. We use the notation $D_{i,j}$ to indicate the smallest (optimal) subset of B_i that uniquely color all the elements of R_j. *In case there are multiple subsets $B'_i \subset B_i$ that uniquely color all the elements of R_j, the subsets B'_i are lexicographically ordered and for $D_{i,j}$, we store the subset B'_i that appears last in the lexicographical ordering.* We build a $m \times n$ table row by row and at the end of the completion of the table, the (m, n)-th entry provides the solution to the problem.

If no solution exists for problem space R_j with solution space B_i, we denote it by setting $D_{i,j} = \emptyset$. Prior to computation of $D_{i,j}$, all elements of the table from rows 1 to $i-1$ have been computed. Moreover, all elements in the i-th row from columns 1 to $j-1$ have also been computed. Only $j+1$ of these elements can be candidates for $D_{i,j}$.

- Case 1: $D_{i-1,j}$, as this provides the smallest subset of B_{i-1} that uniquely colors all elements of R_j.
- Case 2: $D_{i-1,j-1} \cup b_i$, if $D_{i-1,j-1} \neq \emptyset$ and b_i uniquely colors r_j. ($D_{i-1,j-1} \neq \emptyset$ provides a non empty subset of B_{i-1} that assigns unique colors to R_{j-1}. Since b_i uniquely colors r_j, $D_{i-1,j-1}$ together with b_i, uniquely colors R_j).
- Case 3: $D_{i,j'}$, where $D_{i,j'}$ uniquely colors R_j, and j' is the value of j'', such that $|D_{i,j'}|$ is the smallest among all $|D_{i,j''}|, 1 \leq j'' < j$. (It may be noted that $D_{i,j'}$ guarantees unique color assignment to $R_{j'}$ with the smallest subset B'_i of B_i. However, it is possible that for some structure of the graph, not only does B'_i assign a unique coloring to $R_{j'}$, but also it assigns a unique coloring to R_j, where $j' < j$. As such, this should be considered as a candidate for $D_{i,j}$.)

It may be noted that while cases 1 and 2 produce one candidate each, case 3 produces 1 candidate out of $j-1$ possible candidates, for $D_{i,j}$.

With discussions above, $D_{i,j}$ can be expressed in the form of the following recurrence relation,

$$D_{i,j} = min(Set_1, Set_2, Set_3) \tag{1}$$

where, Set_1, Set_2, Set_3 are defined as follows:

$$Set_1 = \quad D_{i-1,j}$$

$$Set_2 = min \begin{cases} D_{i-1,j-1} \cup \{b_i\}, & if\ D_{i-1,j-1} \neq \emptyset \text{ and} \\ & b_i \text{ uniquely colors } r_j \\ \emptyset, & \text{otherwise} \end{cases}$$

$$Set_3 = min \begin{cases} min_{\forall j' < j} D_{i,j'}, & if\ D_{i,j'} \text{ assigns unique} \\ & \text{colors to } R_j \\ \emptyset, & \text{otherwise} \end{cases}$$

In the following, we present an algorithm for MDCS computation for points in one dimensional space, which is based on the recurrence relation given in 1. The algorithm builds a $m \times n$ table, where m and n represents the number of blue and red points respectively, by filling out entries of the table row by row and column by column.

Input: A set of red $R = \{r_1, ..r_n\}$ and blue $B = \{b_1, ..b_m\}$ points on a line.
Output: The smallest subset $B' \subseteq B$ that uniquely colors R.
Initialize: $\forall i, j, D_{i,0} = D_{0,j} = \emptyset$;

```
for i = 1 to m do
    for j = 1 to n do
        Set_1 = D[i-1, j];
        if D[i-1, j-1] ≠ ∅ and b_i uniquely colors r_j then
            Set_2 = D[i-1, j-1] ∪ b_i;
        else
            Set_2 = ∅;
        end
        count = 0;
        max = 1000000;
        idx = 0;
        for k = 1 to j - 1 do
            if D[i, k] uniquely colors R_j then
                count = count + |D[i, k]|;
                if |D[i, k]| < max then
                    max = |D[i, k]|;
                    idx = k;
                end
            end
        end
        if count == 0 then
            Set_3 = ∅;
        else
            Set_3 = D[i, idx];
        end
        D_{i,j} = min(Set_1, Set_2, Set3);
    end
end
```

Algorithm 1: MDCS Computation Algorithm

Theorem 5. *Algorithm 1 finds the optimal solution for the MDCS problem for the points in one-dimensional space.*

Proof. $D_{i,j}$ is the smallest subset B'_i of B_i that uniquely colors R_j. In Algorithm 1, $D_{i,j}$ is computed as the minimum of three sets, Set_1, Set_2, Set_3. One important requirement for any candidate (i.e., a subset of B_i) to be considered as $D_{i,j}$ is that, this subset has to uniquely color R_j.

1) The set $D_{i-1,j}$ uniquely colors R_j, using the smallest subset of B_{i-1}. Since B_{i-1} is a subset of B_i, $D_{i-1,j}$ is a potential candidate for $D_{i,j}$. This is captured in Set_1.
2) The set $D_{i-1,j-1}$ uniquely colors R_{j-1} using the smallest subset of B_{i-1}. However, $D_{i-1,j-1}$, in addition to uniquely coloring R_{j-1}, may or may not uniquely color r_j. If $D_{i-1,j-1}$, in addition to uniquely coloring R_{j-1}, uniquely colors r_j, then problem reduces to Set_1. However, if $D_{i-1,j-1}$, does not uniquely color r_j, but augmentation of the set $D_{i-1,j-1}$, with b_i, enables r_j to receive a unique color, then this augmented set becomes a potential candidate for $D_{i,j}$. This is captured in Set_2.

TABLE V
COMPUTATION OF THE ENTRIES OF THE TABLE

Table Entry	Comparison	Result	Color
D_{11}	min$(D_{01}, D_{00} \cup \{b_1\}, D_{10})$ = min $(\emptyset, \{b_1\}, \emptyset)$	$\{b_1\}$	[A, *, *, *, *]
D_{12}	min$(D_{02}, D_{01} \cup \{b_1\}, D_{11})$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{13}	min$(D_{03}, D_{02} \cup \{b_1\}, min(D_{11}, D_{12}))$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{14}	min$(D_{04}, D_{03} \cup \{b_1\}, min(D_{11}, D_{12}, D_{13}))$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{15}	min$(D_{05}, D_{04} \cup \{b_1\}, min(D_{11}, D_{12}, D_{13}, D_{14}))$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{21}	min$(D_{11}, D_{10} \cup \{b_2\}, D_{20})$ = min $(\{b_1\}, \emptyset, \emptyset)$	$\{b_1\}$	[A, *, *, *, *]
D_{22}	min$(D_{12}, D_{11} \cup \{b_2\}, D_{21})$ = min $(\emptyset, \{b_1 b_2\}, \emptyset)$	$\{b_1 b_2\}$	[A, AB, *, *, *]
D_{23}	min$(D_{13}, D_{12} \cup \{b_2\}, min(D_{21}, D_{22}))$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{24}	min$(D_{14}, D_{13} \cup \{b_2\}, min(D_{21}, D_{22}, D_{23}))$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{25}	min$(D_{15}, D_{14} \cup \{b_2\}, min(D_{21}, D_{22}, D_{23}, D_{24}))$ = min $(\emptyset, \emptyset, \emptyset)$	$\emptyset$	[*, *, *, *, *]
D_{31}	min$(D_{21}, D_{20} \cup \{b_3\}, D_{30})$ = min $(\{b_1\}, \emptyset, \emptyset)$	$\{b_1\}$	[A, *, *, *, *]
D_{32}	min$(D_{22}, D_{21} \cup \{b_3\}, D_{31})$ = min $(\{b_1 b_2\}, \emptyset, \emptyset)$	$\{b_1 b_2\}$	[A, AB, *, *, *]
D_{33}	min$(D_{23}, D_{22} \cup \{b_3\}, min(D_{31}, D_{32}))$ = min $(\emptyset, \{b_1 b_2 b_3\}, \emptyset)$	$\{b_1 b_2 b_3\}$	[A, AB, ABC, BC C]
D_{34}	min$(D_{24}, D_{23} \cup \{b_3\}, min(D_{31}, D_{32}, D_{33}))$ = min $(\emptyset, \emptyset, \{b_1 b_2 b_3\})$	$\{b_1 b_2 b_3\}$	[A, AB, ABC, BC, C]
D_{35}	min$(D_{25}, D_{24} \cup \{b_3\}, min(D_{31}, D_{32}, D_{33}, D_{34}))$ = min $(\emptyset, \emptyset, \{b_1 b_2 b_3\})$	$\{b_1 b_2 b_3\}$	[A, AB, ABC, BC, C]

TABLE VI
TABLE ENTRIES FOR FIG. 6

	r_1	r_1r_2	$r_1r_2r_3$	$r_1r_2r_3r_4$	$r_1r_2r_3r_4r_5$
b_1	$\{b_1\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
b_1b_2	$\{b_1\}$	$\{b_1b_2\}$	$\emptyset$	$\emptyset$	$\emptyset$
$b_1b_2b_3$	$\{b_1\}$	$\{b_1b_2\}$	$\{b_1b_2b_3\}$	$\{b_1b_2b_3\}$	$\{b_1b_2b_3\}$

3) For all $j' < j$, the set $D_{i,j'}$ uniquely colors $R_{j'}$, using the smallest subset of B_i. However, $D_{i,j'}$, in addition to uniquely coloring $R_{j'}$, may or may not uniquely color R_j. Among the set of non-empty $D_{i,j'}$ that uniquely colors R_j, $1 \le j' \le j-1$, the one that has the smallest cardinality is a candidate for $D_{i,j}$. This is Set_3.

$D_{i,j}$ is the smallest subset B'_i of B_i that uniquely colors R_j. 1, 2 and 3 correspond to the following scenarios, (i) $b_i \notin B'_i, B'_i \subset B_i$, (ii) $b_i \in B'_i, B'_i \subset B_i$, and (iii) b_i may or may not be $\in B'_i, B'_i \subset B_i$, depending on whether it satisfies the required condition. Accordingly, these three scenarios are exhaustive from which potential candidates for $D_{i,j}$ must emerge. Since the algorithm examines all three sets, it is guaranteed to find the optimal solution. □

Lemma 6. *MDCS problem for Unit Interval Bigraph generated from m blue points and n red points has no solution in case $n \geq 2m$.*

Proof. There will be m intervals corresponding to m blue points. These intervals may be overlapping with each other, creating a set of subintervals, as shown in Fig. 6b. With m blue points, there can be at most $2m-1$ subintervals. If more than one red point appears in one blue subinterval, then the corresponding red nodes will have exactly identical neighborhood of blue nodes. Accordingly, these red nodes will be "twins". As noted earlier, the necessary and sufficient condition for a graph to have an Identifying Code is that the graph should be "twin" free. If the number of red points n is $2m$ or higher, by Pigeon Hole Principle, then at least one of the blue subintervals must have more than one red point. Accordingly, the corresponding graph will have "twins" and MDCS for the graph will not exist. □

Theorem 7. *Complexity of Algorithm 1 is $\mathcal{O}(m^5)$.*

Proof. The Algorithm 1 builds up a $m \times n$ table. The amount of computation involved in filling out (i, j)-th entry of the table is $\mathcal{O}(j)$, as it has to compute the minimum of the entries $D[i, 1], ..., D[i, j-1]$ and check if the red nodes in R_j received a unique color. The computation involved in checking for uniqueness is $\mathcal{O}(n^2)$. Accordingly, the complexity of the algorithm is $\mathcal{O}(mnjn^2)$. Since $j \leq n$ and $n \leq 2m-1$, the complexity of the algorithm is $\mathcal{O}(m^5)$, where m is the number of blue points, i.e., the potential sensor locations. □

Example: Fig. 6 illustrates 5 red points and 3 blue points in one dimension along with the corresponding bipartite graph. The solution to this problem, using Algorithm 1, can be found in the entry $D_{3,5}$ of Table. VI.

Computational process through which the entries of Table. VI was found is shown in Table. V. Algorithm 1 computes $D_{i,j}$ by finding minimum of the three sets Set_1, Set_2, Set_3. It may be noted that, in this example, all three sets (Set_1, Set_2, Set_3) were used for filling out the entries of $D_{i,j}$ for different values of i, j. Specifically, the entries $D_{2,1}, D_{3,1}, D_{3,2}$ were filled using Set_1, $D_{1,1}, D_{2,2}, D_{3,3}$ were filled using Set_2, and $D_{3,4}, D_{3,5}$ were filled using Set_3. The last column of the Table. V shows the color assignments in the five red nodes $r_1, ..., r_5$. For example, corresponding to the entry $D_{1,1}$, when the node b_1 is injected with a color, only the node r_1 receives the color through seepage from b_1. Similarly, corresponding to the entry $D_{3,2}$, when the nodes b_1, b_2 are injected with two distinct colors (denoted by the alphabets A, B), the node r_1 is colored A, whereas r_2 receives AB. The symbol $*$ indicates that the corresponding r point either did not receive a color or was outside the problem space at that stage.

B. Two Dimensional Case

In this subsection, we provide an Integer Linear Program to find the optimal solution for the problem in two dimensions. Given a set of red points $R = \{r_1, r_2, ..., r_n\}$ and a set of blue points $B = \{b_1, b_2, ..., b_m\}$ on a two-dimensional plane, the Integer Linear Program illustrated below, finds the smallest subset $B' \subseteq B$. such that, injection of colors at these

points, assigns a *unique* color to every red point in R, through seepage. It may be recalled that the construction of a graph from a set of points, has been described in Section. IV.
Instance: $G = (V_1 \cup V_2, E)$, an undirected bipartite graph.
Problem: Find the smallest subset $V_2' \subseteq V_2$, such that injection of colors at these nodes, ensures that each node $v_i \in V_1$, receives a unique color (either atomic or composite) through seepage.
We use the notation $N(v_i)$ to denote the neighborhood of v_i, for any $v_i \in V_1 \cup V_2$. Corresponding to each $v_i \in V_2$, we use an indicator variable x_i,

$$x_i = \begin{cases} 1, & \text{if a color is injected at node } v_i, \\ 0, & \text{otherwise} \end{cases}$$

Objective Function: Minimize $\sum_{v_i \in V_2} x_i$

Coloring Constraint: $\sum_{v_i \in N(v_j)} x_i \geq 1, \forall v_j \in V_1$

Unique Coloring Constraint:

$$\sum_{v_i \in \{N(v_j) \bigoplus N(v_k)\}} x_i \geq 1, \forall v_j, v_k \in V_1, v_j \neq v_k$$

$N(v_j) \bigoplus N(v_k)$ denotes the Exclusive-OR of the node sets $N(v_j)$ and $N(v_k)$. It may be noted that the objective function ensures that the fewest number of nodes in V_2 are assigned a color. The Coloring Constraint ensures that every node in V_1 receives at least one color through seepage from the colors injected at nodes in V_2. A consequence of the Coloring Constraint is that, a node in V_1 may receive more than one color through seepage from the colors injected at nodes in V_2. The Unique Coloring Constraint ensures that, for every pair of nodes (v_j, v_k) in V_1, at least one node in the node set $N(v_j) \bigoplus N(v_k) \subseteq V_2$ is injected with a color. This guarantees that v_j and v_k will not receive identical colors through the color seepage from the nodes in V_2.

VI. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the ILP for the 2D MDCS problem. Conventionally, ILPs tend to be computationally expensive. However, we show that for the 2D MDCS problem, the computation times are fairly small, even for a graph with more than 64000 nodes and 155000 edges, it only took slight more than 3 minutes. To simulate the performance, we generated random points on a 2D plane and constructed graphs following the rules illustrated in Section IV. Table VII illustrates the details of the various graphs and the time taken for computation of B'. The results were computed using the GUROBI Optimization package on an Intel i-5 with 32GB RAM.

TABLE VII
2D MDCS COMPUTATION RESULTS

$\|Nodes\|$	$\|Edges\|$	$\|R\|$	$\|B\|$	$\|B'\|$	$Time(s)$
45	105	5	40	3	0.06
89	187	7	82	4	0.60
170	276	14	156	10	1.59
422	1080	10	412	6	2.75
367	272	9	358	5	2.24
882	427	64	818	55	5.68
5927	3655	155	5772	106	25.18
64655	155339	8999	55656	6020	180.45

VII. CONCLUSION

In this paper, we studied sensor placement problem in one and two dimensional spaces for accurate identification of fault locations. We present a polynomial time algorithm for the MDCS problem for Unit Interval Bigraphs and an ILP for the Unit Disc Bigraphs. To the best of our knowledge, this is the first polynomial time algorithm presented for the computation of MDCS for Unit Interval Bigraphs, when their interval representations are given.

REFERENCES

[1] A. B. Noel, et. al. "Structural Health Monitoring Using Wireless Sensor Networks: A Comprehensive Survey", *IEEE Communications Surveys*. Vol. 19, No. 3, 2017.
[2] B. Wang, "Coverage Problems in Sensor Networks: A Survey", *ACM Computing Surveys*, Vol. 43, No. 4, Article 32, October 2011.
[3] M. Cardei, and J. Wu, "Coverage in Wireless Sensor Networks", *Handbook of Sensor Networks*, CRC Press, Chapter 19.
[4] J. Kleinberg, and E. Tardos, "Algorithm Design", Pearson Education.
[5] B. Li, D. Wang, and Y. Ni, "Demo: On the high quality sensor placement for structural health monitoring", *Proc. IEEE INFOCOM*, 2009.
[6] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Sensor placement with multiple objectives for structural health monitoring", *ACM Trans. Sensor Netw.*, Vol. 10, 2014.
[7] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Deploying wireless sensor networks with fault tolerance for structural health monitoring", *IEEE Trans. Comput.*, Vol. 64, 2015.
[8] J. Skulic, A. Gkelias, and K. K. Leung, "Node placement in linear wireless sensor networks", *Proc. Eur. Signal Process.*, 2013.
[9] S. Kim, et. al. "Health Monitoring of Civil Infrastructures Using Wireless Sensor Networks", *IPSN*, ACM, pp 254–263, 2007.
[10] K. Basu, et al. "Health Monitoring of Critical Power System Equipments Using Identifying Codes." International Conference on Critical Information Infrastructures Security. Springer, Cham, 2018.
[11] J. Lynch, and K. Loh, "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring", *Shock and Vibration Digest*, Vol. 38, pp 91–130, 2006.
[12] A Naeini, and J. Lynch, "Cyber-enabled Wireless Monitoring Systems for Critical Infrastructure Systems", *NIST Civ. Infra. Showcase*, 2014.
[13] M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin, "On a New Class of Codes for Identifying Vertices in Graphs," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, Mar. 1998.
[14] F. Foucaud, Personal Communication, 2018.
[15] N. Bousquet et. al., "Identifying Codes in Hereditary Classes of Graphs and VC-Dimension", arXiv 2017.
[16] M. Laifenfeld, A. Trachtenberg, R. Cohen, and D. Starobinski, "Joint Monitoring and Routing in Wireless Sensor Networks Using Robust Identifying Codes", *Mobile Networks and Applications*, vol. 14, no. 4, pp. 415-432, Aug. 2009.
[17] S. Ray, R. Ungrangsi, F. De. Pellegrini, A. Trachtenburg, and D. Starobinski, "Robust Location Detection in Emergency Sensor Networks", *INFOCOM*, Vol. 2, 2003.
[18] A. Sen, V. H. Goliber, C. Zhou, and K. Basu, "Terrorist Network Monitoring with Identifying Code." *In International Conference on SBP-BRiMS*, pp. 329-339. Springer, Cham, 2018.
[19] F. Foucaud, "Decision and Approximation Complexity for Identifying Codes and Locating-Dominating Sets in Restricted Graph Classes", *Journal of Discrete Algorithms*, Vol. 31, pp 48–68, 2015.
[20] I. Charon, G. Cohen, O. Hudry, and A. Lobstein, "Discriminating Codes in (Bipartite) Planar Graphs", *European Journal of Combinatorics*, Vol. 29, pp 1353 – 1364, 2008.
[21] H. Muller, "Recognizing Interval Digraphs and Interval Bigraphs in Polynomial Time", *Discrete Applied Mathematics*, pp 189–205, 1997.