Sensor Network Design for Uniquely Identifying Sources of Contamination in Water Distribution Networks

Kaustav Basu
School of Computing and Augmented Intelligence
Tempe, USA
kaustav.basu@asu.edu

Arunabha Sen
School of Computing and Augmented Intelligence
Tempe, USA
asen@asu.edu

ABSTRACT

Sensors are being extensively adopted for use in smart cities in order to monitor various parameters, so that any anomalous behaviours manifesting in the deployment area, can be easily detected. Sensors in a deployment area have two functions, sensing/coverage and communication, with this paper focusing on the former. Over the years, several coverage models have been proposed which utilizes the Set Cover based problem formulation. This formulation unfortunately has a drawback, in the sense that it lacks unique identification capability for the location where anomalous behavior is sensed. This limitation can be overcome through utilization of Identifying Code. The optimal solution of the Identifying Code problem provides the minimum number of sensors that will be needed to uniquely identify the location where anomalous behavior is sensed. In this paper, we introduce a novel budget constrained version of the problem, whose goal is to find the largest number of locations that can be uniquely identified with the sensors that can be deployed within the specified budget. We provide an Integer Linear Programming formulation and a Maximum Set-Group Cover (MSGC) formulation for the problem and prove that the MSGC problem cannot have a polynomial time approximation algorithm with a 1/k factor performance guarantee unless P = NP.

CCS CONCEPTS

• **Theory of computation → Graph algorithms analysis**;

KEYWORDS

Smart Cities, Water Distribution Network, Identifying Code, Budget Constrained Identifying Code, Sensor Placement, Maximum Set Cover, Maximum Set-Group Cover

ACM Reference Format:
Kaustav Basu and Arunabha Sen. 2022. Sensor Network Design for Uniquely Identifying Sources of Contamination in Water Distribution Networks. In *Companion Proceedings of the Web Conference 2022 (WWW '22 Companion), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487553.3524850

WWW '22 Companion, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9130-6/22/04…$15.00
https://doi.org/10.1145/3487553.3524850

1 INTRODUCTION

Sensors are used extensively for monitoring various parameters so that any anomalous behaviour can easily be detected [3, 10, 19, 23]. Sensors in a deployment area have two functions, (i) sensing/coverage of target parameters 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. Over the years several coverage models have been proposed, where the underlying assumption is that a sensor placed in a certain location, can sense its environment up to a certain distance. In other words, the sensors have a specific *sensing range* associated with it. This assumption often leads to a Set Cover based problem formulation [22, 31], which unfortunately has a serious limitation, in the sense that it lacks unique identification capability for the location where anomalous behavior is sensed. We have elaborated this limitation in detail in [3]. This point can be summarized in the following way. Assume that a set of sensors, $\mathcal{S} = \{S_1, ..., S_n\}$, is placed in a deployment area to monitor a set of *Points-of-Interest*, $P = \{p_1, ..., p_m\}$. Suppose that a subset $P' \subseteq P$ is within the sensing range of a specific sensor $S_i, S_i \in \mathcal{S}$. In this situation, if any point $p_j \in P'$ starts behaving anomalously, this behavior will be registered by sensor S_i. However, S_i will not be able to determine whether the misbehaving device is at point p_j or any other point p_k, as long as p_k is also within the sensing range of S_i, i.e., $p_k \in P'$. This limitation also exists in the max k-cover problem, a variation of Set Cover.

This limitation can be overcome through utilization of Identifying Code (IC). The optimal solution of the IC problem provides the minimum number of sensors that will be needed to uniquely identify the location where anomalous behavior is sensed. In this paper, we introduce a *budget constrained* version of the problem, whose goal is to find the largest number of locations that can be uniquely identified with the sensors that can be deployed *within the specified budget*. To the best of our knowledge, the budget constrained version of the IC problem has not been studied before. We provide an Integer Linear Programming formulation and a Maximum Set-Group Cover (MSGC) formulation for the problem. We prove that the MSGC problem cannot have a polynomial time approximation algorithm with a 1/k factor performance guarantee unless P = NP.

Our experiments focuses on detecting sources of contaminants in Water Distribution Networks (WDNs). We chose WDNs because of (i) its obvious importance as a critical infrastructure to smart cities; (ii) it has been extensively studied by other researchers [12, 15, 19, 21]; and (iii) synthetic and real WDN graphs are readily available in the public domain [24] for conducting experiments. Prior works have primarily utilized budgeted sensor placement approaches (max k-cover) for the detection of contaminants in such

networks [19]. In this paper, we take this one step further and propose an approach which can not only detect contaminants but also identify its sources, utilizing the concept of ICs.

2 RELATED WORK

The coverage aspect of sensor networks alone has been studied extensively [11, 31]. The survey on Coverage Problems in Sensor Networks [31], references close to 200 papers. Thus, 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 [31]. A more recent survey [30] lists additional studies on the topic. Cardei and Wu in [11], 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 2D or 3D space, has to be monitored. Oftentimes, there are restrictions on the locations (in 2D/3D space), where the sensors can be deployed (cost, terrain, natural elements, etc.), thereby reducing the number of available placement locations.

Sensor coverage problems have a deployment area, where the Points of Interest (PoI) are located and the sensors have to be deployed. A deployment area can be thought of as an (infinite) set of points in a two or three dimensional space. If R, P and Q denote the (infinite) set of points in the deployment area, PoIs in the deployment area and potential locations for sensor placement respectively, then four different case scenarios can be considered, (i) $R = P = Q$, (ii) $R = P$ and $Q \subset R$, (iii) $P = Q$ and $P, Q \subset R$, and (iv) $P \neq Q$ and $P, Q \subset R$. The cases 1 and 2 are considered as *Area Coverage problem* whereas cases 3 and 4 are considered as *Point Coverage problem* (according to [11]), which is the focus of this paper.

The most frequent studied problem in this context is the Sensor Placement Optimization, whose goal is to find the smallest set of locations to deploy sensors, so that all the points of interest can be monitored. If a Boolean disc coverage model [31] is used for sensor coverage, the Sensor Placement Optimization problem can be formulated as a Set Cover problem [18] and a number of studies using this model are available in the literature [11, 31]. Sensor placement for the detection of anomalies in smart cities has been previously studied in [14, 26], where the latter work describes technologies utilized in sensing locations of interest in smart cities (with a case study on a water distribution network). However, as pointed out in Section 1, the Set Cover based approach has a serious limitation which can be overcome by utilizing Identifying Code.

Identifying Codes (IC) was introduced by Karpovsky *et al.* in [16] and other researchers have followed up by studying it both from a theoretical and applicative perspective [13, 27]. Laifenfeld *et al.* studied joint monitoring and routing in wireless sensor networks with IC in [20]. Ray *et al.* studied location detection problem in emergency sensor networks, using IC [27] and presented an algorithm for generating *irreducible* IC in polynomial time. Note that irreducible IC is only a *minimal* IC and may not be the *minimum* (or optimal) IC. Basu *et al.* presented Integer Linear Programs for the computation of the minimum IC set in [4–6, 8, 9, 28] for problems arising from multiple domains, varying from drug/terrorist network monitoring to monitoring critical infrastructures. Sengupta *et al.* utilized Moving Target Defense and IC to present an approach which prevented cyber attacks on sensors placed in critical infrastructures in [29]. The problem of identifying misinformation spreaders in social networks was studied in [2, 7]. Padhee *et al.* utilized ICs for identifying vulnerable assets in critical power systems in [25].

While the goal of all these studies was to find the *minimum* number of sensors to monitor all PoIs, we consider a *budget constrained version*, where the number of sensors that can be deployed is *limited by a pre-approved budget*. Our aim is to *maximize* the number of PoIs that can be monitored with the limited number of sensors that can be procured and deployed. Moreover, we show how our approach can help with the monitoring of Smart Cities, as technologies continue to develop for the same [1].

3 OVERVIEW OF IDENTIFYING CODES

Identifying Code in its simplest form is defined as follows:

Definition 3.1. A vertex set V' of an undirected 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*.

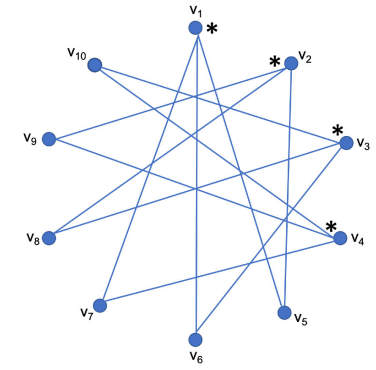

Figure 1: Graph with Identifying Code Set $\{v_1, v_2, v_3, v_4\}$

Table 1: $N^+[v] \cap V'$ results for all $v \in V$ for the graph in Fig. 1

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

The vertices of the set V' may be viewed as *alphabets* of the code, and the *string* made up by the concatenation of the alphabets of $N^+[v] \cap V'$, may be viewed as the unique "code" (or signature) for the node v. For instance, consider the graph $G = (V, E)$ shown in Fig. 1. In this graph $V' = \{v_1, v_2, v_3, v_4\}$ is an ICS, as it can be seen from Table 1 that $N^+[v] \cap V'$ is *unique* for all $v \in V$. 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. *The necessary and sufficient condition*

for a graph to have an Identifying Code is that $G = (V, E)$ be "twin"-free. Two nodes $u, v \in V$ are said to be "twins" if $N^+[v] = N^+[u]$. The definition of the MICS problem for a directed graph is similar to that of the undirected graph and is defined as follows:

Definition 3.2. A vertex set V' of a directed graph $G = (V, E)$ is defined as the Identifying Code Set (ICS) for the vertex set V, if for all $v \in V$, $N^{out}[v] \cap V'$ is unique where, $N^{out}[v] = v \cup N^{out}(v)$ and $N^{out}(v)$ represents the set of out-neighbors of v in $G = (V, E)$. As before, the *Minimum Identifying Code Set* (MICS) problem is to find the Identifying Code Set of *smallest cardinality*.

From Points on the Plane to Graphs:

In Section 1, we described the sensor placement problem in terms of two sets of points in a plane, whereas the MICS problem is described in terms of a graph. From the set of points $\mathcal{S}$ and P we can construct a graph using the following construction rules: (i) For each point $S_i \in \mathcal{S}$ and each point $p_i \in P$, we create node in the graph. Thus the node set V in the $G = (V, E)$, is $V_{\mathcal{S}} \cup V_P$. If a point $p_i \in P$ is within the sensing range of a sensor S_j, then there is an edge in the graph connecting the nodes v_{p_i} and v_{S_j}, where v_{p_i} and v_{S_j} are the nodes corresponding to the points p_i and S_j respectively.

Budget Constrained Identifying Code Set (BCICS) Problem: Given a graph $G = (V, E)$ and an integer B, a subset $V' \subseteq V$ of cardinality at most B is called the Budget Constrained Identifying Code Set of $G = (V, E)$, if it *maximizes* $|V''|$, where $V'' \subseteq V$ and for no two nodes v_i, v_j, such that $v_i \in V''$ and $v_j \in V, N^+(v_i) \cap V' \neq N^+(v_j) \cap V'$.

It may be recalled that the main objective of this study is to deploy the sensors that can be procured within the specified budget, as judiciously as possible, so as to *maximize* the number of locations (points of interest) that can have unique fault identification signature. Moreover, 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.

4 SOLUTIONS FOR BCICS PROBLEMS

In this section, we first show that BCICS can be set up as a generalization of the well studied *Maximum Set Cover* (MSC) problem [17], referred in this paper as *Maximum Set-Group Cover* (MSGC) problem. Next, we present an optimal solution to the BCICS problem.

4.1 Maximum Set-Group Cover Formulation of BCICS Problem

For convenience, we first state the MSC and then the MSGC problems. Then, we explain with the help of an example how the generalization of MSC to MSGC can be used to solve the BCICS problem.

Definition 4.1. Set Cover (SC) Problem: Given a set $A = \{a_1, ..., a_n\}$ and $\mathcal{A}' = \{A'_1, ..., A'_m\}$ ($A'_i \subseteq A, 1 \leq i \leq m$), find the smallest subset $\mathcal{A}'' \subseteq \mathcal{A}'$ such that all the elements in the set A is covered, i,e., every element of A belong to at least one member of $\mathcal{A}''$.

Definition 4.2. Maximum Set Cover (MSC) Problem: Given a set $A = \{a_1, ..., a_n\}$ and subsets $\mathcal{A}' = \{A'_1, ..., A'_m\}$ ($A'_i \subseteq A, 1 \leq i \leq m$) and an integer B, find the largest subset $A'' \subseteq A$ that can be covered by using a subset $\mathcal{A}''' \subseteq \mathcal{A}'$, where $|\mathcal{A}'''| \leq B$.

Definition 4.3. Maximum Set-Group Cover (MSGC) Problem: Given a set $A = \{a_1, ..., a_n\}$ and subsets $\mathcal{A}' = \{A'_1, ..., A'_m\}$ ($A'_i \subseteq A, 1 \leq i \leq m$) and $\mathcal{G} = \{G_1, ..., G_p\}$ ($G_i \subseteq A, 1 \leq i \leq p$) and an integer B, find the subset $A'' \subseteq \mathcal{A}'$ with $|A''| = B$ that maximizes the number of *groups* completely "covered" by A'', i.e., it finds the largest cardinality subset $G' \subseteq G$ that satisfies the condition that $\forall G_j \in G', \cup_{A'_i \in A''} A'_i \cap G_j = G_j$.

We elaborate the formulation of the BCICS problem as a MSGC problem with the help of the example shown in Fig. 1. We introduce a few definitions before the explanation.

Definition 4.4. Closed Neighborhood of $v_i = CN(v_i) = N^+(v_i)$, where $N^+(v_i) = N(v_i) \cup \{v\}$, and $N(v_i)$ is the set of nodes adjacent to v_i.

Definition 4.5. Distinguishing Set for v_i and $v_j = DS(v_i, v_j) = CN(v_i) \bigoplus CN(v_j)$, where $\bigoplus$ denotes *Exclusive-OR* operation. At least one element of the set must be selected to distinguish between the nodes v_i and v_j.

Definition 4.6. Isolation Set for $v_i = IS(v_i) = \cup_{j=1}^{n} \{v_{i_j} : v_{i_j} \in DS(v_i, v_j)\}, j \neq i$. This is the set of sets, such that if all nodes in a set is selected, it will distinguish (isolate/uniquely identify) v_i from all other nodes of the graph.

Definition 4.7. Presence Set for v_i, $PS(v_i) = \{CN(v_j) : v_i \in CN(v_j)\} \cup \{DS(v_j, v_k) : v_i \in DS(v_j, v_k)\}, \forall v_i, v_j, v_k, 1 \leq v_i, v_j, v_k \leq n$. $PS(v_i)$ is the set of all $CN(v_j)$s and $DS(v_i, v_j)$s, where v_i is present.

The number of nodes in the graph in Fig. 1 is 10, i.e., $n = 10$. Accordingly, there will be 10 $CN(v_i)$ sets and corresponding to each v_i, there will be 9 $DS(v_i, v_j)$ sets, $(\forall v_i, v_j \neq v_i)$. Hence, there will be 100 sets altogether. However, it may be noted that these 100 sets will not be distinct, as $DS(v_i, v_j) = DS(v_j, v_i)$. Thus, the total number of distinct $DS(v_i, v_j)$ sets in this example will be $\sum_{i=1}^{n-1} i = 45$ (as $n = 10$). These 10 $CN(v_i)$ and 45 $DS(v_i, v_j)$ sets are shown in Table 2, and are marked as a_1 through a_{55}. The Presence Sets for nodes 1 through 10 for the example graph, are shown in Table 3. The Isolation Sets for nodes 1 through 10 for the example graph, are shown in Table 4.

The BCICS problems can be viewed as an MSGC problem in the following way. We say $PS(v_i)$ "hits" $CN(v_j)$ if $PS(v_i) \cap CN(v_j) \neq \emptyset$. Similarly, $PS(v_i)$ "hits" $DS(v_j, v_k)$ if $PS(v_i) \cap DS(v_j, v_k) \neq \emptyset$. With slight misuse of the language, we use the term "cover" instead of "hit", i.e., we will say $PS(v_i)$ "covers" $CN(v_j)$ if $PS(v_i) \cap CN(v_j) \neq \emptyset$ and "covers" $DS(v_j, v_k)$ if $PS(v_i) \cap DS(v_j, v_k) \neq \emptyset$. In Table 2, the $CN(v_j)$ and $DS(v_j, v_k)$ sets are numbered from a_1 through a_{10} and a_{11} through a_{55} respectively ($A = \{a_1, \ldots, a_{55}\}$) . Each $PS(v_i)$ is a subset of the set A, and is denoted as $A'(v_i)$ in Table 3. We define the set $A' = \{A'_1, \ldots, A'_{10}\}$. From Tables 2 and 3, it can be seen that $A'(1) = PS(1)$ covers $a_1 = CN(1), a_5 = CN(9), a_{37} = DS(4, 7)$ and 25 other $CN(v_i)$ or $DS(v_j, v_k)$ sets shown in the first row of Table 3. The set G is defined as the $IS(v_i), 1 \leq i \leq 10$, i.e., $G = \{G_1, \ldots, G_{10}\}$. Hence, the BCICS problem can be formulated as a MSGC problem.

It may be noted that the MSC problem is a generalization of the SC problem and the MSGC problem is a generalization of the MSC problem. As SC is a well known NP-complete problem [18],

Table 2: $CN(v_i)$ and $DS(v_i, v_j)$ Table for all $i, j,\ 1 \leq i, j \leq n$; $A = \{a_1, \ldots, a_{55}\}$

$a_1 = CN(1) = \{1, 5, 6, 7\}$	$a_2 = CN(2) = \{2, 5, 8, 9\}$	$a_3 = CN(3) = \{3, 6, 8, 10\}$	$a_4 = CN(4) = \{4, 7, 9, 10\}$
$a_5 = CN(5) = \{1, 2, 5\}$	$a_6 = CN(6) = \{1, 3, 6\}$	$a_7 = CN(7) = \{1, 4, 7\}$	$a_8 = CN(8) = \{2, 3, 8\}$
$a_9 = CN(9) = \{2, 4, 9\}$	$a_{10} = CN(10) = \{3, 4, 10\}$	$a_{11} = DS(1, 2) = \{1, 2, 6, 7, 8, 9\}$	$a_{12} = DS(1, 3) = \{1, 3, 5, 7, 8, 10\}$
$a_{13} = DS(1, 4) = \{1, 4, 5, 6, 9, 10\}$	$a_{14} = DS(1, 5) = \{2, 6, 7\}$	$a_{15} = DS(1, 6) = \{3, 5, 7\}$	$a_{16} = DS(1, 7) = \{4, 5, 6\}$
$a_{17} = DS(1, 8) = \{1, 2, 3, 5, 6, 7, 8\}$	$a_{18} = DS(1, 9) = \{1, 2, 4, 5, 6, 7, 9\}$	$a_{19} = DS(1, 10) = \{1, 3, 4, 5, 6, 7, 10\}$	$a_{20} = DS(2, 3) = \{2, 3, 5, 6, 9, 10\}$
$a_{21} = DS(2, 4) = \{2, 4, 5, 7, 8, 10\}$	$a_{22} = DS(2, 5) = \{1, 8, 9\}$	$a_{23} = DS(2, 6) = \{1, 2, 3, 5, 6, 8, 9\}$	$a_{24} = DS(2, 7) = \{1, 2, 4, 5, 7, 8, 9\}$
$a_{25} = DS(2, 8) = \{3, 5, 9\}$	$a_{26} = DS(2, 9) = \{4, 5, 8\}$	$a_{27} = DS(2, 10) = \{2, 3, 4, 5, 8, 9, 10\}$	$a_{28} = DS(3, 4) = \{3, 4, 6, 7, 8, 9\}$
$a_{29} = DS(3, 5) = \{1, 2, 3, 5, 6, 8, 10\}$	$a_{30} = DS(3, 6) = \{1, 8, 10\}$	$a_{31} = DS(3, 7) = \{1, 3, 4, 6, 7, 8, 10\}$	$a_{32} = DS(3, 8) = \{10, 2, 6\}$
$a_{33} = DS(3, 9) = \{2, 3, 4, 6, 8, 9, 10\}$	$a_{34} = DS(3, 10) = \{4, 6, 8\}$	$a_{35} = DS(4, 5) = \{1, 2, 4, 5, 7, 9, 10\}$	$a_{36} = DS(4, 6) = \{1, 3, 4, 6, 7, 9, 10\}$
$a_{37} = DS(4, 7) = \{1, 9, 10\}$	$a_{38} = DS(4, 8) = \{2, 3, 4, 7, 8, 9, 10\}$	$a_{39} = DS(4, 9) = \{2, 7, 10\}$	$a_{40} = DS(4, 10) = \{3, 7, 9\}$
$a_{41} = DS(5, 6) = \{2, 3, 5, 6\}$	$a_{42} = DS(5, 7) = \{2, 4, 5, 7\}$	$a_{43} = DS(5, 8) = \{1, 3, 5, 8\}$	$a_{44} = DS(5, 9) = \{1, 4, 5, 9\}$
$a_{45} = DS(5, 10) = \{1, 2, 3, 4, 5, 10\}$	$a_{46} = DS(6, 7) = \{3, 4, 6, 7\}$	$a_{47} = DS(6, 8) = \{1, 2, 6, 8\}$	$a_{48} = DS(6, 9) = \{1, 2, 3, 4, 6, 9\}$
$a_{49} = DS(6, 10) = \{1, 4, 6, 10\}$	$a_{50} = DS(7, 8) = \{1, 2, 3, 4, 7, 8\}$	$a_{51} = DS(7, 9) = \{1, 2, 7, 9\}$	$a_{52} = DS(7, 10) = \{1, 3, 7, 10\}$
$a_{53} = DS(8, 9) = \{3, 4, 8, 9\}$	$a_{54} = DS(8, 10) = \{2, 4, 8, 10\}$	$a_{55} = DS(9, 10) = \{2, 3, 9, 10\}$	

Table 3: $PS(v_i)$ Table for all $i,\ 1 \leq i \leq n$

$A'_1 = PS(1) = \{a_1, a_5, a_6, a_7, a_{11}, a_{12}, a_{13}, a_{17}, a_{18}, a_{19}, a_{22}, a_{23}, a_{24}, a_{29}, a_{30}, a_{31}, a_{35}, a_{36}, a_{37}, a_{43}, a_{44}, a_{45}, a_{47}, a_{48}, a_{49}, a_{50}, a_{51}, a_{52}\}$
$A'_2 = PS(2) = \{a_2, a_5, a_8, a_9, a_{11}, a_{14}, a_{17}, a_{18}, a_{20}, a_{21}, a_{23}, a_{24}, a_{27}, a_{29}, a_{32}, a_{33}, a_{35}, a_{38}, a_{39}, a_{41}, a_{42}, a_{45}, a_{47}, a_{48}, a_{50}, a_{51}, a_{54}, a_{55}\}$
$A'_3 = PS(3) = \{a_3, a_6, a_8, a_{10}, a_{12}, a_{15}, a_{17}, a_{19}, a_{20}, a_{23}, a_{25}, a_{27}, a_{28}, a_{29}, a_{31}, a_{33}, a_{36}, a_{38}, a_{40}, a_{41}, a_{43}, a_{45}, a_{46}, a_{48}, a_{50}, a_{52}, a_{53}, a_{55}\}$
$A'_4 = PS(4) = \{a_4, a_7, a_9, a_{10}, a_{13}, a_{16}, a_{18}, a_{19}, a_{21}, a_{24}, a_{26}, a_{27}, a_{28}, a_{31}, a_{33}, a_{34}, a_{35}, a_{36}, a_{38}, a_{42}, a_{44}, a_{45}, a_{46}, a_{48}, a_{49}, a_{50}, a_{53}, a_{54}\}$
$A'_5 = PS(5) = \{a_1, a_2, a_5, a_{12}, a_{13}, a_{15}, a_{16}, a_{17}, a_{18}, a_{19}, a_{20}, a_{21}, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}, a_{29}, a_{35}, a_{41}, a_{42}, a_{43}, a_{44}, a_{45}\}$
$A'_6 = PS(6) = \{a_1, a_3, a_6, a_{11}, a_{13}, a_{14}, a_{16}, a_{17}, a_{18}, a_{19}, a_{20}, a_{23}, a_{28}, a_{29}, a_{31}, a_{32}, a_{33}, a_{34}, a_{36}, a_{41}, a_{46}, a_{47}, a_{48}, a_{49}\}$
$A'_7 = PS(7) = \{a_1, a_4, a_7, a_{11}, a_{12}, a_{14}, a_{15}, a_{17}, a_{18}, a_{19}, a_{21}, a_{24}, a_{28}, a_{31}, a_{35}, a_{36}, a_{38}, a_{39}, a_{40}, a_{42}, a_{46}, a_{50}, a_{51}, a_{52}\}$
$A'_8 = PS(8) = \{a_2, a_3, a_8, a_{11}, a_{12}, a_{17}, a_{21}, a_{22}, a_{23}, a_{24}, a_{26}, a_{27}, a_{28}, a_{29}, a_{30}, a_{31}, a_{33}, a_{34}, a_{38}, a_{43}, a_{47}, a_{50}, a_{53}, a_{54}\}$
$A'_9 = PS(9) = \{a_2, a_4, a_9, a_{11}, a_{13}, a_{18}, a_{20}, a_{22}, a_{23}, a_{24}, a_{25}, a_{27}, a_{28}, a_{33}, a_{35}, a_{36}, a_{37}, a_{38}, a_{40}, a_{44}, a_{48}, a_{51}, a_{53}, a_{55}\}$
$A'_{10} = PS(10) = \{a_3, a_4, a_{10}, a_{12}, a_{13}, a_{19}, a_{20}, a_{21}, a_{27}, a_{29}, a_{30}, a_{31}, a_{32}, a_{33}, a_{35}, a_{36}, a_{37}, a_{38}, a_{39}, a_{45}, a_{49}, a_{52}, a_{54}, a_{55}\}$

Table 4: $IS(v_i)$ Table for all $i,\ 1 \leq i \leq n$

$G_1 = IS(1) = \{a_1, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, a_{16}, a_{17}, a_{18}, a_{19}\}$
$G_2 = IS(2) = \{a_2, a_{11}, a_{20}, a_{21}, a_{22}, a_{23}, a_{24}, a_{25}, a_{26}, a_{27}\}$
$G_3 = IS(3) = \{a_3, a_{12}, a_{20}, a_{28}, a_{29}, a_{30}, a_{31}, a_{32}, a_{33}, a_{34}\}$
$G_4 = IS(4) = \{a_4, a_{13}, a_{21}, a_{28}, a_{35}, a_{36}, a_{37}, a_{38}, a_{39}, a_{40}\}$
$G_5 = IS(5) = \{a_5, a_{14}, a_{22}, a_{29}, a_{35}, a_{41}, a_{42}, a_{43}, a_{44}, a_{45}\}$
$G_6 = IS(6) = \{a_6, a_{15}, a_{23}, a_{30}, a_{36}, a_{41}, a_{46}, a_{47}, a_{48}, a_{49}\}$
$G_7 = IS(7) = \{a_7, a_{16}, a_{24}, a_{31}, a_{37}, a_{42}, a_{46}, a_{50}, a_{51}, a_{52}\}$
$G_8 = IS(8) = \{a_8, a_{17}, a_{25}, a_{32}, a_{38}, a_{43}, a_{47}, a_{50}, a_{53}, a_{54}\}$
$G_9 = IS(9) = \{a_9, a_{18}, a_{26}, a_{33}, a_{39}, a_{44}, a_{48}, a_{51}, a_{53}, a_{55}\}$
$G_{10} = IS(10) = \{a_{10}, a_{19}, a_{27}, a_{34}, a_{40}, a_{45}, a_{49}, a_{52}, a_{54}, a_{55}\}$

it can easily verified that both MSC and MSGC problems are NP-Complete. However, unlike the SC problem for which a *log n* factor approximation algorithm exists, and for the MSC problem for which a $(1 - 1/e)$ factor approximation algorithm exists, in the following, we show that $1/k$ factor approximation algorithm ($k > 1$) for the MSGC problem cannot exist unless $P = NP$.

Theorem 4.8. *Unless P = NP, there cannot be a polynomial time approximation algorithm for the MSGC problem with a performance factor that guarantees for every instance I of the MSGC problem $APP(I) \geq \lceil OPT(I)/k \rceil$, where $APP(I)$ and $OPT(I)$ represents the approximate and optimal solutions respectively for the MSGC problem instance I and k is a real number with $k > 1$.*

Proof: We claim that if such an algorithm existed, the Set Cover problem, which is known to be NP-complete, could have been solved in polynomial time. Suppose if possible, such an algorithm APP_{MSGC} exists. From an instance of the SC problem, given as $A_{SC} = \{a_1, ..., a_n\}$ and $\mathcal{A}'_{SC} = \{A'_1, ..., A'_m\}$ ($A'_i \subseteq A_{SC}, 1 \leq i \leq m$), we create an instance of the MSGC problem, by making n copies of the instance of the SC problem. Thus,

$$A_{MSGC} = \{a_1^1, \ldots, a_n^1, a_1^2, \ldots, a_n^2, \ldots, a_1^n, \ldots, a_n^n\} \quad (1)$$

$$\mathcal{A}'_{\mathcal{MSGC}} = \{A'^1_1, ..., A'^n_1, A'^1_2, ..., A'^n_2, \ldots, A'^1_m, ..., A'^m_n\} \quad (2)$$

$(A'^i_j \subseteq A_{MSGC}, 1 \leq i, j \leq n), A'^i_j = \{a^i_k | a_k \in A_j, \forall i, 1 \leq i \leq n, 1 \leq j \leq m\}$, $\mathcal{G}_{\mathcal{MSGC}} = \{G_1^{MSGC}, ..., G_n^{MSGC}\}$, and $G_i^{MSGC} = \{a_1^i, \ldots, a_n^i\}, \forall i, 1 \leq i \leq n)$. ($G_i^{MSGC} \subseteq A_{MSGC}, 1 \leq i \leq n$).

Table 5: Example of creation of an instance of Maximum Set-Group Cover Problem from an instance of Set Cover Problem

Set Cover	Maximum Set Group Cover
1. $A_{SC} = \{a_1, a_2, a_3\}$	1. $A_{MSGC} = \{a_1^1, a_2^1, a_3^1, a_1^2, a_2^2, a_3^2, a_1^3, a_2^3, a_3^3\}$
2. $\mathcal{A}' = \{A_1, A_2\}$	2. $\mathcal{A}' = \{A_1^1, A_1^2, A_1^3, A_2^1, A_2^2, A_2^3\}$
3. $A_1 = \{a_1, a_2\}$	3. $A_1^1 = \{a_1^1, a_2^1\}, A_1^2 = \{a_1^2, a_2^2\}, A_1^3 = \{a_1^3, a_2^3\}$
4. $A_2 = \{a_2, a_3\}$	4. $A_2^1 = \{a_2^1, a_3^1\}, A_2^2 = \{a_2^2, a_3^2\}, A_2^3 = \{a_2^3, a_3^3\}$
	5. $G = \{G_1, G_2, G_3\}, G_1 = \{a_1^1, a_2^1, a_3^1\}, G_2 = \{a_1^2, a_2^2, a_3^2\}, G_3 = \{a_1^3, a_2^3, a_3^3\}$

An example of construction of an instance of the MSGC problem from an instance of the SC problem is shown in Table 5. Given an instance of the SC problem, using the MSGC instance creation rules above, we can create the corresponding instance of the MSGC problem. If there is a polynomial time algorithm APP_{MSGC} with $APP(I) \geq \lceil OPT(I)/k) \rceil$, performance guarantee, we can apply it to the instance of the MSGC problem created from the instance of the SC problem. The algorithm will either return zero, implying that no group can be completely covered, or a non-zero number, implying

that at least one group can be completely covered. If the algorithm returns zero, we can conclude that the SC problem has no solution. If the algorithm returns a non-zero number, it implies that the SC problem has a solution. Thus, we can conclude that if there exists a polynomial time approximation algorithm for the MSGC problem with a performance guarantee of $APP(I) \geq \lceil OPT(I)/k \rceil$, for some real number k, $k \geq 1$, then the SC problem, which is known to be NP-complete, can be solved in polynomial time. This implies that unless $P = NP$, no such polynomial time approximation algorithm can exist for the MSGC problem.

4.2 Optimal Solution for the BCICS Problem with ILP

The goal of BCICS is to have unique signature for as many nodes as possible, subject to the constraint that the number of nodes where sensor is placed does not exceed the specified budget, B.

Instance: A graph $G = (V, E)$ and an integer B.
Problem: Find a subset $V' \subseteq V$ of cardinality B (i.e., $|V'| = B$ such that placement of sensors at these nodes ensures that a largest subset of nodes V'' of V has a unique signature associated with it.

For each $v_i \in V$, we use an indicator variable x_i, such that

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

Also, for each $v_i \in V$, we use an indicator variable y_i, such that

$$y_j = \begin{cases} 1, & \text{if a } v_j \text{ ends up having a unique signature,} \\ 0, & \text{otherwise} \end{cases}$$

Objective Function: Maximize $\sum_{v_j \in V} y_j$

Budget Constraint: $\sum_{v_i \in V} x_i \leq B$,

In addition to the Budget Constraint, we introduce two additional constraints, *Coverage Constraint* and Unique Coverage Constraint. Before we introduce the constraints, we first define the terms *Coverage* and *Unique Coverage.*

Definition 4.9. Coverage of a node v_i is the *Closed Neighborhood Set* of node v_i, and is denoted by $Cov(v_i) = \{v_i \cup N^+(v_i)\}$.

Definition 4.10. Unique Coverage of a node pair (v_i, v_j) is the *Exclusive-OR* of the Closed Neighborhood Set of the nodes v_i and v_j and is denoted by $Uni\_Cov(v_i, v_j) = N^+(v_j) \bigoplus N^+(v_k)$

The Coverage and the Unique Coverage constraints are:

Coverage Constraint: $\forall v_j \in V$

$M_1 \times (1 - y_j) + \sum_{v_i \in N^+(v_j)} x_i \geq 1$,

Unique Coverage Constraint: $\forall v_j, v_k \in V, v_j \neq v_k$

$M_2 \times (2 - y_j - y_k) + \sum_{v_i \in \{N^+(v_j) \bigoplus N^+(v_k)\}} x_i \geq 1$

Note that the objective function ensures that the largest number of nodes in V receives a unique signature. The budget constraint ensures that not more than B nodes in V can be selected for sensor placement. The Coverage Constraint ensures that if node v_i has a unique signature (i.e., $y_i = 1$), a sensor must be placed in at least one node in its closed neighborhood (as otherwise v_i will not have any signature, let alone a unique signature). The Unique Coverage Constraint ensures that for every pair of nodes (v_j, v_k) in V to have unique signatures associated with them, (i.e., $y_j = 1$ and $y_k = 1$), a sensor must have been placed in at least one node in the node set $N^+(v_j) \bigoplus N^+(v_k)$. This guarantees that v_j and v_k will not have identical signatures. The parameters M_1 and M_2 in the constraints are two large constants.

5 EXPERIMENTAL RESULTS

The datasets used in our experimentation were obtained from the Kentucky Water Resources Research Institute [24]. Each dataset contains information regarding the collection of pipes, pumps, valves, junctions, tanks, and reservoirs that make up a water distribution system. For our study, the points of interest are the nodes which represent junctions, tanks, and reservoirs whereas the edges represent pipes, pumps, and valves.

The results of our experimentation on the datasets are presented in Table 6. For the ease of understanding, we will describe in detail the result for the first row, i.e., the Fourteen Pipes dataset. This dataset has 12 nodes and 26 edges. The third column denotes the MICS cardinality for this graph. The rationale for computing the MICS for the datasets was to ensure that we varied the budget parameter according to the MICS cardinality. In our setup, we have considered three different budget parameters, $k = 25\%, 50\%$ and 75% of the MICS cardinality. For instance, in the Fourteen Pipes dataset, the MICS solution is 8 and the budget parameters are $k = 25\%$ of $8 = 2$, $k = 50\%$ of $8 = 4$ and $k = 75\%$ of $8 = 6$. Now, the task is to identify the locations for sensor placement which maximizes the number of nodes which would be uniquely covered (i.e., will have a unique fault signature). For the Fourteen Pipes dataset, with at most 2 sensors ($k = 25\%$), we see that the optimal number of nodes uniquely covered is 3, for which the coverage % (% of nodes uniquely covered with the budget) is $\frac{3}{12} = 25\%$, where 12 is the number of nodes in the network. Similarly, for $k = 50\%$ or when the budget is 4, we can uniquely cover 7 nodes optimally, with 58.33% coverage, and finally for $k = 75\%$, we can cover 10 nodes optimally, with 83.33% coverage. Similar results for the other datasets follow. Note that, for all the datasets considered, the coverage $\% \geq k$. In other words, the benefits, in terms of the number of nodes uniquely covered, outweighs the cost (budget). The average benefits for the costs $k = 25\%, 50\%, 75\%$ are 30.96%, 58.04% and 82.42%.

Fig. 2 illustrates the time taken by the optimal approach. Note that the optimal solution computes the solution for the largest graph considered fairly quickly, i.e., in a couple of minutes.

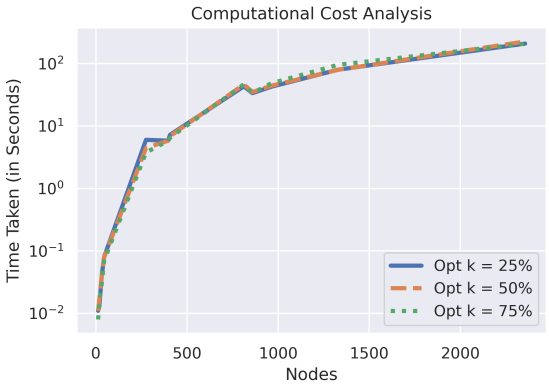

Figure 2: BCICS Computational Run Time

Table 6: Optimal BCICS Results for Water Distribution Network Systems

Dataset	Num Nodes	Num Edges	MICS Solution	k = 25%		k = 50%		k = 75%	
				OPT	Cov. %	OPT	Cov. %	OPT	Cov. %
Fourteen Pipes	12	26	8	3	25%	7	58.33%	10	83.33%
Modified 19 Pipe	14	35	9	6	42.85%	9	64.28%	12	85.71%
Hanoi	32	66	21	10	31.25%	18	56.25%	25	78.12%
FOWM	45	94	30	14	31.11%	25	55.55%	37	82.22%
Kentucky 3	275	646	161	83	30.18%	160	58.18%	227	82.54%
Calibration	396	840	257	119	30.05%	228	57.57%	327	82.57%
Long Term	407	866	263	121	29.73%	234	57.49%	336	82.55%
Kentucky 2	812	1927	485	250	30.78%	476	58.62%	672	82.75%
Kentucky 1	859	1844	548	257	29.91%	496	57.74%	710	82.65%
Kentucky 4	962	2103	619	294	30.56%	561	58.31%	795	82.64%
Kentucky 8	1329	2936	826	412	31%	786	59.14%	1113	83.74%
Kentucky 12	2355	4810	1583	686	29.12%	1296	55.03%	1890	80.25%

6 CONCLUSION

In this short paper we introduced a novel budget constrained version of the Identifying Code problem, geared towards identifying sources of anomalies in water distribution systems of smart cities. We provided an optimal solution for the problem through ILP and proved that no approximate algorithm for the MSGC with $1/k$ factor bound ($k \geq 1$) can exist, unless $P = NP$. Conventionally, ILPs tend to be computationally expensive, however, in our experimentation, computation times are fairly small, even for graphs with more than 2300 nodes and 4800 edges. It took less than a couple of minutes using GUROBI on an Intel i-9 processor with 128GB RAM.

REFERENCES

[1] Leonidas G Anthopoulos. 2017. The smart city in practice. In *Understanding Smart Cities: A Tool for Smart Government or an Industrial Trick?* Springer, 47–185.

[2] Kaustav Basu. 2019. Identification of the source (s) of misinformation propagation utilizing identifying codes. In *Companion Proceedings of The 2019 World Wide Web Conference*. 7–11.

[3] Kaustav Basu, Sanjana Dey, Subhas Nandy, and Arunabha Sen. 2019. Sensor networks for structural health monitoring of critical infrastructures using identifying codes. In *2019 15th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, 43–50.

[4] Kaustav Basu, Malhar Padhee, Sohini Roy, Anamitra Pal, Arunabha Sen, Matthew Rhodes, and Brian Keel. 2018. Health monitoring of critical power system equipments using identifying codes. In *International Conference on Critical Information Infrastructures Security*. Springer, 29–41.

[5] Kaustav Basu and Arunabha Sen. 2019. Monitoring individuals in drug trafficking organizations: A social network analysis. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 480–483.

[6] Kaustav Basu and Arunabha Sen. 2019. On augmented identifying codes for monitoring drug trafficking organizations. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 1136–1139.

[7] Kaustav Basu and Arunabha Sen. 2021. Epidemiological Model Independent Misinformation Source Identification. (2021).

[8] Kaustav Basu and Arunabha Sen. 2021. Identifying individuals associated with organized criminal networks: a social network analysis. *Social Networks* 64 (2021), 42–54.

[9] Kaustav Basu, Chenyang Zhou, Arunabha Sen, and Victoria Horan Goliber. 2018. A novel graph analytic approach to monitor terrorist networks. In *2018 IEEE International Conference on Social Computing & Networking (SocialCom)*. IEEE, 1159–1166.

[10] Md Zakirul Alam Bhuiyan, Guojun Wang, Jiannong Cao, and Jie Wu. 2014. Sensor placement with multiple objectives for structural health monitoring. *ACM Transactions on Sensor Networks (TOSN)* 10, 4 (2014), 1–45.

[11] Mihaela Cardei and Jie Wu. 2004. Coverage in wireless sensor networks. *Handbook of sensor networks* 21 (2004), 201–202.

[12] Demetrios G Eliades and Marios M Polycarpou. 2009. A fault diagnosis and security framework for water systems. *IEEE Transactions on Control Systems Technology* 18, 6 (2009), 1254–1265.

[13] Florent Foucaud. 2015. Decision and approximation complexity for identifying codes and locating-dominating sets in restricted graph classes. *Journal of discrete algorithms* 31 (2015), 48–68.

[14] Gerhard P Hancke, Gerhard P Hancke Jr, et al. 2013. The role of advanced sensing in smart cities. *Sensors* 13, 1 (2013), 393–425.

[15] William E Hart and Regan Murray. 2010. Review of sensor placement strategies for contamination warning systems in drinking water distribution systems. *Journal of Water Resources Planning and Management* 136, 6 (2010), 611–619.

[16] Mark G Karpovsky, Krishnendu Chakrabarty, and Lev B Levitin. 1998. On a new class of codes for identifying vertices in graphs. *IEEE transactions on information theory* 44, 2 (1998), 599–611.

[17] Samir Khuller, Anna Moss, and Joseph Seffi Naor. 1999. The budgeted maximum coverage problem. *Information processing letters* 70, 1 (1999), 39–45.

[18] Jon Kleinberg and Eva Tardos. 2006. *Algorithm design*. Pearson Education.

[19] Andreas Krause, Jure Leskovec, Carlos Guestrin, Jeanne VanBriesen, and Christos Faloutsos. 2008. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management* 134, 6 (2008), 516–526.

[20] Moshe Laifenfeld, Ari Trachtenberg, Reuven Cohen, and David Starobinski. 2009. Joint monitoring and routing in wireless sensor networks using robust identifying codes. *Mobile networks and applications* 14, 4 (2009), 415–432.

[21] Byoung Ho Lee and Rolf A Deininger. 1992. Optimal locations of monitoring stations in water distribution system. *Journal of Environmental Engineering* 118, 1 (1992), 4–16.

[22] Dieyan Liang, Hong Shen, and Lin Chen. 2021. Maximum Target Coverage Problem in Mobile Wireless Sensor Networks. *Sensors* 21, 1 (2021), 184.

[23] Adam B. Noel, Abderrazak Abdaoui, Tarek Elfouly, Mohamed Hossam Ahmed, Ahmed Badawy, and Mohamed S. Shehata. 2017. Structural Health Monitoring Using Wireless Sensor Networks: A Comprehensive Survey. *IEEE Communications Surveys Tutorials* 19, 3 (2017), 1403–1423. https://doi.org/10.1109/COMST.2017.2691551

[24] University of Kentucky. 2001. Kentucky Water Resources Research Institute. Retrieved January 30, 2022 from https://uknowledge.uky.edu/kwrri/

[25] Malhar Padhee, Reetam Sen Biswas, Anamitra Pal, Kaustav Basu, and Arunabha Sen. 2020. Identifying unique power system signatures for determining vulnerability of critical power system assets. *ACM SIGMETRICS Performance Evaluation Review* 47, 4 (2020), 8–11.

[26] Nordine Quadar, Abdellah Chehri, Gwanggil Jeon, and Awais Ahmad. 2021. Smart water distribution system based on IoT networks, a critical review. *Human Centred Intelligent Systems* (2021), 293–303.

[27] Saikat Ray, Rachanee Ungrangsi, De Pellegrini, Ari Trachtenberg, and David Starobinski. 2003. Robust location detection in emergency sensor networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, Vol. 2. IEEE, 1044–1053.

[28] Arunabha Sen, Victoria Horan Goliber, Chenyang Zhou, and Kaustav Basu. 2018. Terrorist Network Monitoring with Identifying Code. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 329–339.

[29] Sailik Sengupta, Kaustav Basu, Arunabha Sen, and Subbarao Kambhampati. 2020. Moving target defense for robust monitoring of electric grid transformers in adversarial environments. In *International Conference on Decision and Game Theory for Security*. Springer, 241–253.

[30] Abhishek Tripathi, Hari Prabhat Gupta, Tanima Dutta, Rahul Mishra, KK Shukla, and Satyabrat Jit. 2018. Coverage and connectivity in WSNs: A survey, research issues and challenges. *IEEE Access* 6 (2018), 26971–26992.

[31] Bang Wang. 2011. Coverage problems in sensor networks: A survey. *ACM Computing Surveys (CSUR)* 43, 4 (2011), 1–53.