An Exploration of Exact Methods for Effective Network Failure Detection and Diagnosis

Auguste Burlats, Pierre Schaus, and Cristel Pelsser

ICTeam, UCLouvain, Belgium
firstname.surname@uclouvain.be

Abstract. In computer networks, swift recovery from failures requires prompt detection and diagnosis. Protocols such as Bidirectional Forwarding Detection (BFD) exists to probe the liveliness of a path and endpoint. These protocols are run on specific nodes that are designated as network monitors. Monitors are responsible for continuously verifying the viability of communication paths. It is important to carefully select monitors as monitoring incurs a cost, necessitating finding a balance between the number of monitor nodes and the monitoring quality. Here, we examine two monitoring challenges from the Boolean network tomography research field: coverage, which involves detecting failures, and 1-identifiability, which additionally requires identifying the failing link or node. We show that minimizing the number of monitors while meeting these requirements constitutes NP-hard problems. We present integer linear programming (ILP), constraint programming (CP) and MaxSAT formulations for these problems and compare their performance. Using 625 real network topologies, we demonstrate that employing such exact methods can reduce the number of monitors needed compared to the existing state-of-the-art greedy algorithm.

Keywords: Integer linear programming · Constraint Programming · MaxSAT · Boolean tomography · Network supervision.

1 Introduction

Computer networks form the backbone of modern digital communication, and their reliability is crucial for maintaining seamless connectivity across various sectors. Failures within these networks can have significant consequences, leading to service disruption and potential financial loss. As such, it is essential to develop efficient and accurate methods for detecting and diagnosing network failures, enabling swift recovery and minimizing the impact on end-users.

Various protocols enable to monitor the liveliness of Internet paths from the well-known `ping` utility present in most operating systems, and used by measurement infrastructures such as RIPE Atlas [23], to more recent protocols such as BFD [15]. In addition to fasten link failure detection when deployed on adjacent routers, BFD enables to quickly detect failures along a path, such ability being leveraged in Software Defined Networks (SDN) to quickly detect and report failures to a network controller.

In this study, we focus on Boolean network tomography, a research field that holds great promise for enhancing the resilience of networks. Boolean network tomography combines end-to-end measures, performed with ping or BFD, for example, with inference algorithms to estimate the state of different elements in the network. Its advantage is that it only requires a subset of nodes to be monitors and supervise an entire network. With this approach, monitors send messages to each other through *measurement paths*. When a failure occurs on a node, all paths that cross it fail. Thus, the failure can be detected by observing if some measurement paths are not working. If the set of failed measurement paths forms a unique signature, then it is even possible to identify the failed node.

In the remaining of the paper, we treat the case of node failures. Indeed, we can easily account for edge-failure considerations by transforming the network graph. Adding dummy nodes to represent each link, enables to transform the node failure detection, respectively localization, problem, into detecting and locating link failures, applying the approach in this paper.

Our investigation focuses on minimizing the number of designated monitor nodes while ensuring some level of quality of network monitoring. This is crucial for minimizing monitoring costs without compromising the network's overall health and performance. We explore two critical monitoring challenges: the *cover problem*, which seeks to detect failures, and the *1-identifiability problem*, which requires pinpointing the exact failing node. Figure 1 illustrates these concepts.

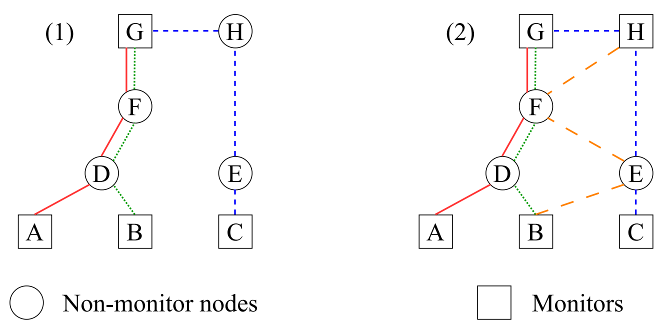

Fig. 1. Illustration of 1-identifiability: (1) All nodes are covered but not 1-identifiable. (2) All nodes are 1-identifiable. Each color represents a path. In the situation (1), the path linking nodes B and H is not a measurement path because H is not a monitor. Each node is covered because they are all crossed by a path linking two monitors, but if D or F fails, in each case the paths between A and G and between B and G will both fail. The failure will thus be detected, but it would not be possible to know which of the two nodes is the origin of the failure. The problem is the same with nodes E and H. In the situation (2), H is a monitor, thus the path linking B and H is a measurement path (in dashed orange). Therefore each non-monitor node is now crossed by a unique set of paths : If D fails, paths (A,G) and (B,G) will fail; if E fails it will be paths (C,G) and (B,H); if F is not functional, paths (A,G), (B,G) and (B,H) will not work. By looking at which paths are not working versus the alive paths, it is possible to infer without ambiguity which node is the origin of the breakdown.

Conceptually, a node failure in a network results in the disruption of all paths traversing it. These affected paths collectively constitute the *symptom* associated with the failing node. A network is covered if there is a non-empty symptom for each node. Additionally, a network is considered 1-identifiable if every node possesses a unique, non-empty symptom, thereby serving as an identifier for the node in the event of a failure. By compiling a comprehensive list of these identifiers, one can efficiently diagnose a failure by simply observing the disrupted paths and cross-referencing a precomputed table that maps the failed paths (symptoms) to the corresponding node.

The *1-identifiability* problem can be generalized as the *k-identifiability* problem for the localization of up to k simultaneous failures. Most precomputed failure protection mechanisms today are tailored for single faults. With BFD and our proposal, locating a single failure enables to use such protections without waiting for the convergence of the routing protocol, speeding up recovery. However, in case of multiple failures, the protections for recovery are usually not in place. Hence, in this situation we are forced to wait for the routing protocol to converge. We have thus limited benefits of quickly identifying k faults. In this paper, we focus on selecting the smallest set of monitors ensuring that the entire network is either covered or 1-identifiable. But our approach can theoretically be extended to k-identifiability.

An important assumption of the considered networks in this study is that the routes between any pairs of nodes are imposed by the routing protocol and known by the planning tool that will select the monitors. A pair of monitors is only able to verify the status of those routes. In practice network operators usually configure link (IGP) weights to influence where the traffic flows in the network assuming they follow shortest paths (see for instance [5] for optimizing IGP weights). Alternatively, other protocols such as segment routing or MPLS [8, 9, 17] make it possible to introduce deviations or explicit route set-ups between pairs of nodes, deviating from shortest paths. For all these protocols, the monitors are able to determine which data paths between them are affected by a failure.

Our contributions can be summarized as follows.

- We formulate the optimal monitor placement problem for the cover and 1-identifiability (Section 3);
- We demonstrate that the node cover and 1-identifiability problems are NP-hard (Section 4);
- We introduce an integer linear programming (ILP), a constraint programming (CP) and a MaxSAT model to resolve this problem. We propose redundant constraints and reductions to shrink the search space (Section 5);
- We also introduce a specialized version of the MNMP greedy algorithm [19], tailored for 1-identifiability, to compare our exact methodologies with this approach (Section 6).
- We evaluate their performance through a comparative analysis using 625 real network topologies (Section 7).

Our findings reveal that the introduced models can reduce the amount of monitor nodes compared to a greedy approach while maintaining coverage or 1-identifiability, paving the way for more robust and reliable telecommunication networks.

2 Related works

Various methods have been investigated to address the monitor placement issue for failure detection, beyond utilizing exact methods. For example, the Maximum Node-identifiability Monitor Placement (MNMP) [19] is a greedy algorithm that progressively adds monitors to achieve the desired k-identifiability and subsequently removes any unnecessary monitors. Unfortunately, it does not guarantee optimality for 1-identifiability problem. Bezerra et al. [2] suggest various improvements to this algorithm to decrease computational cost and enable its use in wireless networks that experience more frequent changes. Our approach here is different, as we use exact methods that allow us to reduce the number of selected monitors, as shown in Section 7, and also because we are focusing on coverage and 1-identifiability.

Stanic et al. [25] present an ILP model and a greedy algorithm for the monitor placement for fault localization in transparent all-optical networks, which is an analogous problem to ours. The main difference being that, in their context, each measurement path requires only one monitor. Their model could be applicable here by having monitors probing non-monitor nodes and waiting for their answer. However, it requires to assume that the routes are symmetrical which is rare in practice [14, 12, 26]. Here we consider routes that are not necessarily symmetrical; for them to be measurement paths, both of their two ends must be monitors.

The dual version of the problem considered in this paper, where the number of monitors is limited and the number of identifiable nodes or links needs to be maximized has also been studied. Ren et al. [22] design a greedy algorithm that chooses monitors such as the number of k-identifiable links is maximized. Bartolini et al. provide in [1] an upper bound for the maximum number of identifiable nodes given a specific measurement path budget. Ma et al. propose the Greedy Maximal identifiability Monitor Placement [18], an algorithm that incrementally adds the monitors that maximize the number of identifiable links, until the maximal budget of monitors is reached. Here, the definition of identifiable is extended to all types of additive metrics (delays, packet delivery ratios, ...) and not only the failure detection.

Related problems, involving tomography for monitoring, have also been examined. He et al. [10] investigate, in the context of Network Function Virtualization (NFV), the challenge of positioning services in a network with a predetermined set of measurement paths to optimize their identifiability while ensuring a high Quality of Service (QoS). Zhang et al. [27] adapt network tomography techniques to supervise traffic in smart cities, which includes determining the optimal placement for monitoring cameras. A possible approach to reduce the load of Boolean network tomography is to partition the network in multiple areas and to locate failures in each area independently. Ogino et al. [20] offer a procedure to divide the network in such areas and a scheme to manage them. This partitioning can be run before the monitor selection, then, instead of dealing with a single large instance of the problem considered in this paper, there would be multiple smaller ones. We refer to chapters 5-6 of the book [11] for a complete review of Boolean-Network Tomography.

3 Problem formulation

The following two paragraphs formally introduce the problems addressed in this paper, specifically the monitor cover and the 1-identifiability problems.

The Monitor Cover Problem Assuming that the network topology is known, it can be represented as a connected graph. We further assume that a given route exists between every pair of nodes, but only the ones linking monitor nodes are considered measurement paths. The symptom of a node is defined as the collection of measurement paths passing through it. The objective of this problem is to identify the minimum number of monitors from the network's node set such that every node is traversed by a minimum of one measurement path. The problem can be formalized as follows. We consider an oriented graph $G = (V, E)$ where there is a cycle-free route between each pair of nodes $(i, j) \in V^2$. In this problem the order in which nodes are crossed by a route does not matter, we thus represent routes as unordered sets containing all the nodes crossed by these routes and we denote $P(i, j) = \{i, \ldots, j\}$ the route linking i and j. Routes are not necessarily symmetric; thus $P(i, j)$ and $P(j, i)$ don't necessarily contain the same nodes. The goal is to find the minimal set of monitors $M \subseteq V$ such as $\cup_{(i,j) \in M^2} P(i, j) = V$.

The Monitor 1-identifiability Problem This problem adds one constraint over the cover problem. If one node fails, we don't only need the failure to be detectable, but we also aim to be able to locate it without ambiguity. For the failure to be identifiable, the symptom of the failure (set of failed measurement paths) must be unique. The problem can be formalized as follows. We consider the same oriented graph $G = (V, E)$ and set of routes P as for the monitor cover problem. For a set of monitors M, we denote by S_i the symptom of node i: $S_i = \{(i', j') \in M^2 \mid i \in P(i', j')\}$. The goal is to find the minimal set of monitors $M \subseteq V$ such that $\cup_{(i,j) \in M^2} P(i, j) = V$ (cover) and $\forall i \neq j \in V^2, S_i \neq S_j$ (1-identifiability).

4 Complexity of optimal monitor placement

In this section we study the complexity of our two problems and show that they are both NP-hard.

Theorem 1. *The monitor cover problem is NP-hard.*

Proof. We reduce the set cover problem to the monitor cover problem as follows. Consider an instance of the set cover problem defined by $(S = \{S_1, \ldots, S_k\}, U)$, where S is a set of sets and U is the universe. For the monitor cover problem, we construct a set of vertices $V = S \cup U \cup \{T\}$, including one vertex for each set in S, one for each element in U, and a special vertex T, designated as a monitor in every valid solution.

There is an edge between every pair of nodes (i, j) in V^2. The set of routes is defined as follows. For each pair of nodes (i, j) in U^2, the route between them is direct, i.e., $P(i, j) = \{i, j\}$ if $i \neq j$. In addition to these routes, for each $S_i \in S$, an arbitrary route between S_i and T contains all nodes in S_i. Finally, a route $P(T, T)$ arbitrarily passes through all nodes in S. This set of routes ensures that in any optimal solution, only nodes in S plus the node T are selected as monitors.

First, observe that T necessarily needs to be a monitor, as it is only traversed by routes for which it is the origin or the destination. The simple selection of T also ensures the coverage of all the nodes in S. Then, choosing a node $S_i \in S$ as a monitor to cover a node $U_j \in U$ is at least as cost-effective as choosing U_j itself as a monitor. Since we ensured by construction that choosing a set vertex $S_i \in S$ as a monitor covers all nodes in S_i, the optimal solution to the set cover problem can be retrieved from the set of nodes in S designated as monitors. □

Example 1. As an example, consider a set cover problem where $U = \{1, 2, 3, 4, 5\}$ and the collection of sets is $S = \{\{1, 2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$. The corresponding vertex set is $V = S \cup U \cup \{T\} = \{\{1, 2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}, 1, 2, 3, 4, 5, T\}$. In our example, the path $P(\{1, 2, 3\}, T)$ includes nodes 1, 2, and 3 in an arbitrary order and finishes at T. Similar paths are built from the other nodes in S. The path $P(T, T)$ includes $\{1, 2, 3\}$, $\{2, 4\}$, $\{3, 4\}$, and $\{4, 5\}$ in an arbitrary order. It is easy to see that selecting $\{1, 2, 3\}$ as a monitor is more cost-effective than selecting 1, 2, and 3 individually.

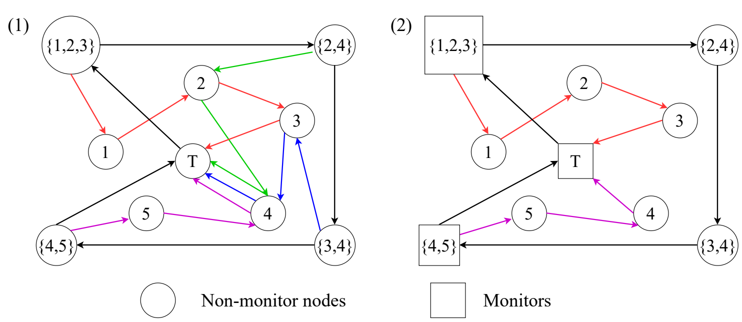

Fig. 2. Illustration of a monitor cover problem. (1) shows the instance with all major paths (2-nodes paths and edges are not shown for readability). (2) shows the optimal solution.

Theorem 2. *The monitor 1-identifiability problem is NP-hard.*

Proof. We reduce the monitor cover problem (known to be NP-hard from theorem 1) to the monitor 1-identifiability problem. Consider a monitor cover problem instance $(G = (V, E), P)$, where G is the graph and P is the set of routes. We construct a monitor 1-identifiability problem $(G^* = (V^*, E^*), P^*)$ as follows. The set of vertices is $V^* = V \cup V' \cup \{T\}$ where V is the set of vertices of the monitor cover problem, V' is a set of companions vertices for V (each vertex $i \in V$ has its companion $i' \in V'$, thus $|V| = |V'|$) and T is a special vertex that is linked to each node in V. Nodes in V' will be designated as monitors in every valid solution. The set of edges E^* is composed of the edges in $E \cup \{(i, T), \forall i \in V\} \cup \{(i, i') \in V \times V'$ such as i' is the companion of $i\}$.

The routes connecting each pair of nodes $(i, j) \in V^2$ are inherited from the monitor cover problem, i.e., $P \subset P^*$. The other routes are represented by the sets in $P' = P^* \backslash P$. For all nodes in V', the route linking them is defined as $P'(i', j') = \{i', i, T, j, j'\}\ \forall (i', j') \in V'^2$. Every other routes in P' are segments of those routes, e.g., the route linking a node $i \in V$ to a companion node $j' \in V'$ of another node $j \in V$ is defined as $P'(i, j') = P'(j', i) = \{i, T, j, j'\}$, or the route linking T to a companion node $i' \in V$ is $P'(i', T) = P'(T, i') = \{i', i, T\}$.

First, note that every node in V' is necessarily a monitor, as they are only present at the extremities of routes. Their selection allows each node in $V*$ to be covered. T is also 1-identifiable as it is the only node crossed by each of these routes. The nodes in V are distinguishable from every other node in the exception of their companion node. Choosing T as a monitor is not effective as the only routes that cross nodes in V without necessarily crossing their companions (and thus make them 1-identifiable) are the routes imported from the cover problem. We ensured by construction that selecting a pair of node $i, j \in V^2$ enables 1-identifiability for every node covered by $P(i, j)$, hence the optimal solution to the monitor cover problem can be retrieved from the nodes in V designated as monitors in the 1-identifiability problem. □

Example 2. As an example, consider a monitor cover problem where $V = \{A, B, C, D, E\}$ and $E = \{(A, B), (B, C), (C, D), (A, D), (C, E)\}$. Figure 3 presents the corresponding topology in the 1-identifiability problem. A', B', C', D' and E' are necessarily monitors, T is thus covered by all the paths connecting them and has a unique symptom. A' and A are both covered by every path starting or ending from A' $(P'(A', B'), P'(B', A'), \ldots)$. Their symptoms are equal but different from each other node. Every node in V is in a similar situation, distinguishable from all the other nodes except their companion. If $P(D, B) = \{D, A, B\}$, selecting D and B as monitors allows covering A, B and D in the cover problem. In the 1-identifiability problem, it makes them 1-identifiability because $P(D, B)$ does not contain A', B' and D'.

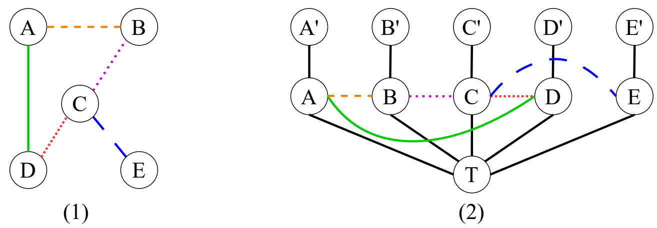

Fig. 3. Illustration of the transformation of a cover problem to an 1-identifiability problem. The routes are not represented. (1) shows the graph for the monitor cover problem. (2) shows the corresponding graph for the monitor 1-identifiability problem, edges from the graph (1) are identifiable by their colors.

5 Models for Optimal Monitor Placement

This section gives the models for the resolution of cover and 1-identifiability problems. We first present the ILP, CP and MaxSAT models. Then we introduce some reductions and redundant constraints of the problem to tighten the formulation.

5.1 Models Definition

Integer Linear Programming Model The problem is modeled with two binary variable vectors : x is a vector of size $|V|$ modeling the set of monitors (x_i is true iff node i is a monitor) and y is a vector of size $|P|$ modeling the set of measurement paths ($y_{P(i,j)}$ is true iff the route $P(i,j)$ is a measurement path). We denote by S_i the subset of P containing all the routes crossing node i; we also denote by $D_{i,j}$ the set of routes crossing node i without going by j and vice versa, $D_{i,j} = (S_i \cup S_j) - (S_i \cap S_j)$. The ILP model is composed of the following constraints:

$$\text{minimize} \sum_{i \in V} x_i \tag{1}$$

subject to:

$$y_{P(i,j)} \leq x_i \qquad \forall P(i,j) \in P \tag{2}$$

$$y_{P(i,j)} \leq x_j \qquad \forall P(i,j) \in P \tag{3}$$

$$y_{P(i,j)} \geq x_i + x_j - 1 \qquad \forall P(i,j) \in P \tag{4}$$

$$\sum_{P(i,j) \in S_{i'}} y_{P(i,j)} \geq 1 \qquad \forall i' \in V \tag{5}$$

$$\sum_{P(i,j) \in D_{i',j'}} y_{P(i,j)} \geq 1 \qquad \forall i', j' \in V^2 \tag{6}$$

Equation (1) expresses the objective function that is to minimize the number of monitors. Equations (2)-(4) link the monitor selection to the measurement path variables. A route $P(i,j)$ is considered a measurement path ($y_{P(i,j)} =$

1) iff both its starting and ending nodes are monitors ($x_i = 1$ and $x_j = 1$). Equation (5) models the cover constraints. For a node to be covered at least one route among S_i needs to be a measurement path. Thus, for each node i we have the constraint (5). Finally, equation (6) models the 1-identifiability constraints. A node is 1-identifiable iff it is distinguishable of each other nodes. Constraining each node to be 1-identifiable is equivalent to constraint each pair of nodes to be distinguishable. For two nodes to be distinguishable, it is sufficient that at least one measurement path crosses one of the nodes without crossing the other. For the 1-identifiability problem, it is important to keep the cover constraints from (5), otherwise a solution with one uncovered node would be a valid solution, as its symptom would be different from each other symptom. Notice that the constraint (6) is defined for pairs of nodes, but the same idea can be applied to all combinations of up to k nodes to model the k-identifiability constraints. Unfortunately, the number of constraints grows exponentially with k.

Constraint Programming Model The CP model follows the same logic as the ILP model and uses the same set of binary variables x and y. The reified constraints (8) ensure that route $P(i,j)$ is considered a measurement path ($y_{P(i,j)} = 1$) iff both its starting and ending nodes are monitors ($x_i = 1$ and $x_j = 1$). The sum constraints ensuring the coverage and the 1-identifiability are replaced by logical or constraints.

$$\text{minimize} \sum_{i \in V} x_i \tag{7}$$

subject to:

$$y_{P(i,j)} \equiv x_i \wedge x_j \qquad \forall P(i,j) \in P \tag{8}$$

$$\bigvee_{P(i,j) \in S_{i'}} y_{P(i,j)} \qquad \forall i' \in V \tag{9}$$

$$\bigvee_{P(i,j) \in D_{i',j'}} y_{P(i,j)} \qquad \forall i', j' \in V^2 \tag{10}$$

MaxSAT Model The translation of the CP model to a MaxSAT model is straightforward : there are two sets of literals, x and y, that correspond to the binary variables of the CP model. The reified constraints in (8) are translated in three sets of clauses defined by equations (12)-(14). The clauses ensuring the coverage and the 1-identifiability, in equations (15) and (16), are inherited from the *or* constraints. All the clauses defined by the equations (12)-(16) are hard constraints that must be satisfied. To translate the objective function we add one weighted clause for each node $i \in V$. Each one of them only contains one literal, x_i, and their cost is fixed to -1. Hence, to maximize the sum of the satisfied constraints' weights, we need to minimize the number of nodes selected to be monitors.

$$\text{maximize} \sum_{i \in V} -1 * x_i \tag{11}$$

subject to:

$$x_i \vee \neg y_{P(i,j)} \quad \forall P(i,j) \in P \tag{12}$$

$$x_j \vee \neg y_{P(i,j)} \quad \forall P(i,j) \in P \tag{13}$$

$$\neg x_i \vee \neg x_j \vee y_{P(i,j)} \quad \forall P(i,j) \in P \tag{14}$$

$$\bigvee_{P(i,j) \in S_{i'}} y_{P(i,j)} \quad \forall i' \in V \tag{15}$$

$$\bigvee_{P(i,j) \in D_{i',j'}} y_{P(i,j)} \quad \forall i', j' \in V^2 \tag{16}$$

5.2 Problem reductions and Redundant Constraints

To reduce the search space and thus the computation costs, one can add nodes that must be monitors and redundant constraints to obtain a tighter model.

Detecting monitors The nodes that are only covered by paths originating or ending in them must be monitors to be covered. We refer to these nodes as leaf nodes $= \{i \in V \mid \forall i', j' \neq i, i \notin P(i', j')\}$. Clearly, nodes with degree one are part of the set of leaf nodes. This can be observed in the case of node H in Figure 4.

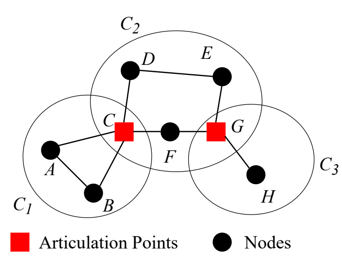

Fig. 4. Illustration of bi-connected components. C_1, C_2 and C_3 represents the 3 bi-connected components. Components C_1 and C_2 are linked together by the articulation node C. Removing this node would result in two connected parts: the first one only composed of C_1 and the other one composed of C_2 and C_3.

Redundant Constraints A bi-connected component of a graph is a subgraph in which every pair of vertices is connected by at least two disjoint paths, meaning that the subgraph remains connected even if any single vertex or edge is removed. These subgraphs are linked together by articulation points, nodes whose removal would disconnect the total graph. For example, figure 4 shows a graph composed of three bi-connected components. The bi-connected components and their articulation points can be identified in linear time with a depth-first search [13]. We denote by $V_C \subseteq V$ the subset of nodes in the bi-connected component C that are not articulation points and $A_C \subseteq V$ the subset of articulation points in C.

Lemma 1. *For a bi-connected component C with exactly one articulation point, ensuring coverage implies that at least one node in V_C must be a monitor.*

Proof. Because routes are cycle-free, a component with only one articulation point can't be crossed by a route connecting two nodes outside of a component. Otherwise, it would require the route to cross the articulation point at least twice, resulting in the presence of a cycle. Thus, it requires at least one monitor in the component to cover its nodes. As an example, in Figure 4 component C_1 contains only one articulation point (the node C). Because routes are cycle-free, it is clear that if A and B are not monitors, then no measurement paths will go through the component. We can thus add a constraint which enforces that at least one node in such a bi-connected component must be a monitor. □

We thus add the following constraints to the ILP model :

$$\sum_{i \in V_C} x_i \geq 1 \qquad \forall\ \texttt{bi-connected}(C) \in G \text{ with } |A_C| = 1 \tag{17}$$

Their equivalents in the CP and the MaxSAT models are:

$$\bigvee_{i \in V_C} x_i \qquad \forall\ \texttt{bi-connected}(C) \in G \text{ with } |A_C| = 1 \tag{18}$$

6 A greedy algorithm for 1-identifiability

To position our models within the state-of-the-art, we compare them with the greedy algorithm MNMP [19] (more specifically the MNMP-UP version). MNMP is a generic algorithm introduced to solve the broader k-identifiability problem. Its complexity is $O(|P|^2 \cdot |V|^3) = O(|V|^7)$, as $|P| = |V|^2$ in our case. We introduce a version dedicated to the 1-identifiability problem, as outlined in Algorithm 1, that reduces the complexity to $O(|V|^5)$. This is the version we use in our experimental comparisons.

6.1 Description of the Algorithm

First, the set of monitors is initialized with the set of leaf nodes (i.e., nodes that only are at extremities of routes) since it is the only way to cover them. Then the first while loop iteratively adds monitors to the solution until each node in the network is covered. During each iteration, the node selected for inclusion is the one covering the maximum number of currently uncovered nodes, considering

Algorithm 1 MNMP Implementation for the Monitor Cover and 1-identifiability Problems

1: $M \leftarrow$ leaf nodes
2: $U \leftarrow V$ ▷ Uncovered Nodes
3: $U' \leftarrow V^2$ ▷ Indistinguishable pairs of Nodes
4: $U \leftarrow U \backslash \Psi(M)$ ▷ Coverage
5: **while** $U \neq \emptyset$ **do**
6: $\quad m \leftarrow \arg\max_{w \in V \backslash M} |U \cap \Psi(M \cup \{w\})|$
7: $\quad U \leftarrow U \backslash \Psi(M \cup \{m\})$
8: $\quad M \leftarrow M \cup \{m\}$
9: **end while**
10: **return** M if goal is coverage
11: $U' \leftarrow U' \backslash \Omega(M)$
12: **while** $U' \neq \emptyset$ **do** ▷ 1-identifiability
13: $\quad m \leftarrow \arg\max_{w \in V \backslash M} |U' \cap \Omega(M \cup \{w\}|$
14: $\quad U' \leftarrow U' \backslash \Omega(M \cup \{m\})$
15: $\quad M \leftarrow M \cup \{m\}$
16: **end while**
17: **return** M

the present set of monitors. In the pseudo-code, we denote by U the set of uncovered nodes and $\Psi(M) \subseteq V$ represents the set of nodes covered by the union of all paths between each pair of nodes in M, i.e., $\Psi(M) = \bigcup_{i,j \in M^2} P_{i,j}$. If the objective is solely to cover the nodes, the algorithm halts at this point. Otherwise, the algorithm proceeds with the second 'while' loop, which is designed to make each node 1-identifiable. Once again, the algorithm adds monitors in an iterative manner. However, this time the criterion for adding monitors is the number of pairs of currently indistinguishable nodes that the new monitor would turn distinguishable, taking into account the current set of monitors. We denote by U' the set of pairs of indistinguishable nodes. $\Omega(M) \subseteq V^2$ contains the pair of nodes that are distinguishable under the union of all paths between each pair of nodes in M. The original MNMP algorithm contains a third loop where it iterates on each monitors and test if it is redundant, i.e., if removing the monitor would impact the coverage or the 1-identifiability. If the monitor is redundant, it is then removed from the solution. However, we observed on our instances that no monitors were removed during this loop. Thus, we removed this last loop.

6.2 Time Complexity

Cover problem Leaf nodes are detected in $O(|V^3|)$. The set of uncovered nodes U and the routes are represented as bit sets of size $|V|$, where the bit i is set if node i is uncovered or if i is in the route. U and $\Psi(M)$ are computed by logical *or* and *and* operations on these bit sets in $O(|V|)$. Therefore, line 4 requires $O(|V| \cdot |M|^2)$ operation. For the monitor cover problem, the complexity is dominated by the loop in lines 5-9. The most computationally demanding step within the loop is line 6: computing $|U \cap \Psi(M \cup \{w\}|$ for all $w \in V \backslash M$ requires to compute for each route linking a monitor to a non-monitor node ($O(|P|)$) the set of uncovered nodes that it crosses (($O(|V|)$). If we assume that $|P| = |V^2|$, the resulting complexity is $O(|V^3|)$. The worst case for the monitor cover problem is when $O(|V|)$ monitors are required to cover the graph. In this case, the while loop takes $O(|V|^4)$. The overall time complexity of MNMP for the monitor cover problem is $O(|V|^4)$.

1-identifiability problem Computing the set of pairs made distinguishable by a route is done in $O(|V|^2)$. Thereby, computing $\Omega(M)$ requires $O(|M|^2 \cdot |V|^2)$ and the reduction of U' in line 11 takes $O(|V|^2 \cdot |M|^2)$. Line 13 is very similar to line 6. It requires to calculate the set of node pairs ($O(|V|^2)$) that can be distinguished by every route between a monitor and a non-monitor node. The resulting complexity is $O(V^4)$. In the worst case (i.e., the first while loop returns $O(1)$ monitors and $O(|V|)$ monitors are required for 1-identifiability) the while loop in lines 12 to 16 takes $O(|V|^5)$. Hence the time complexity of MNMP for the 1-identifiability problem is $O(|V|^5)$ in the worst case.

7 Experimental Results

In this section we evaluate our different approaches.[1]. The ILP model relies on Gurobi [7]. We run the CP model on OR-Tools [21]. For the MaxSAT approach, we use NuWLS-c [4]. Experiences run on a SkyLake CPU with up to 95 GB of memory. Every test is limited to a runtime of 3 minutes and is allocated 20 GB of memory.

Dataset We run our models on topologies from *Rocketfuel* [24], *Internet Topology Zoo* [16] and CAIDA's *ITDK* (IPv4 and from February 2022) [3]. We assume shortest hop-count paths with arbitrary tie breaking when multiple best paths exist. For an even more realistic set of routes, we use topologies from Repetita [6], which contain IGP weights. In non-connected topologies, we keep the largest component. For the largest topologies, the solvers encountered memory issue on the 1-identifiability problem. To ensure a fair comparison between the solvers, we removed the instances for which at least one solver faced memory issues. As a result, for the cover problem we have 625 topologies with up to 960 nodes (the median number of nodes is 33). For the 1-identifiability problem, there are 587 instances with up to 330 nodes (the median being 31).

Results Table 1 displays the number of instances where the returned solution is the best among those obtained by different solvers, as well as the number of instances for which the solvers proved the optimality of their solutions. Since the solvers established the optimality of their solutions in over 99% of the instances, the returned solution often corresponds to the optimal one. To compare solvers when they don't return the best solution, figure 5 shows the cumulative count of instances for which each solver provides a solution below a given number of extra monitors, compared to the optimal.

[1] Source code available at `https://github.com/BurlatsAuguste/MonitorPlacement`

Solver	Goal	Best solution found	Optimality proven
Gurobi	Cover	623 (99.68%)	621 (99.36%)
	1-identifiability	583 (99.32%)	583 (99.32%)
OR-Tools	Cover	**624 (99.84%)**	**624 (99.84%)**
	1-identifiability	**587 (100.0%)**	**585 (99.66%)**
NuWLS-c	Cover	621 (99.36%)	583 (93.28%)
	1-identifiability	577 (98.30%)	566 (96.42%)
MNMP	Cover	590 (94.40%)	None
	1-identifiability	420 (71.55%)	None

Table 1. Number of solved instances for each model

Comparison with MNMP Table 1 shows that the greedy algorithm fails to find the best solution for 6.60% of instances in the monitor cover problem and for 28.45% of instances in the monitor 1-identifiability problem. While all three exact solvers can find the best solution for a greater number of instances, especially for the 1-identifiability problem (more than 93%). This highlights the effectiveness of our models, which have the ability to reduce the number of monitors required in numerous scenarios. As we can see in figure 5, most of the time the improvement represents 1 to 3 monitors. Each new monitor adds $2*|M-1|$ measurement paths that need to be regularly probed. Thus, in large topologies that require many monitors, having some unnecessary monitors can strongly impact the traffic, especially around the monitors, and congestion can occur. In a context where minimizing the number of monitor is crucial, the exact solvers are a pertinent choice, as they are able to offer better solutions than MNMP.

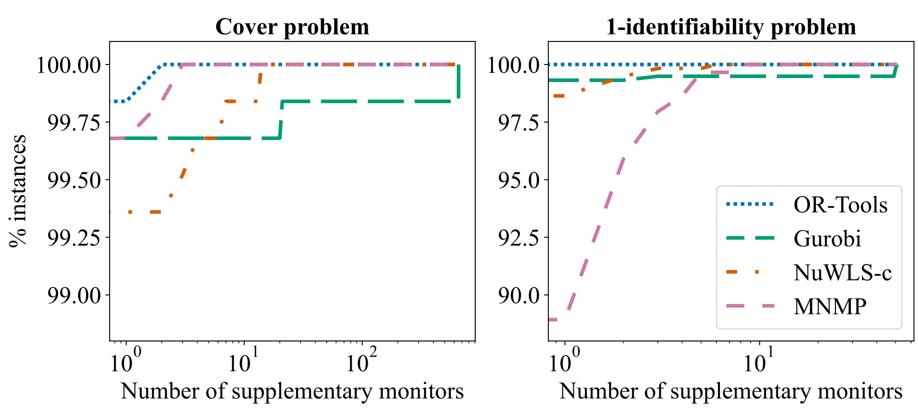

Fig. 5. Proportion of instances solved against the maximal distance on the objective with the best solution found (in number of monitors). Because of the logarithmic scale, the lines start with the proportion of instance for which the solver choose at most one supplementary monitor.

Comparison of the Exact Solvers Among the exacts solvers, OR-Tools seems to be the best approach. It returns the best solution for all the instances in the 1-identifiability problem and for 99.84% in the cover problem. It is also the model that proves the optimality of its solution on the most instances (99.84% for the Cover problem and 99.66% for the 1-identifiability problem). For one particular instance of the cover problem, Gurobi returns a solution with more than 600 supplementary monitors. What happens is that the solver reaches the time limit before the end of Gurobi's presolving. For the 1-identifiability problem, for 3 instances Gurobi returns a solution containing 50 more monitors than the best solution. The size of the instance is not the problem. Indeed, they contain fewer than 100 nodes, when the solver is able to find the optimal solution instances three times larger. Instead the diameter of the graph is determinant here.

8 Conclusion

In this paper, we studied the placement of monitors in a network to ensure coverage or 1-identifiability of all nodes. We demonstrated that the problem is NP-hard for both objectives. We proposed three exact models: an Integer Linear Programming (ILP) model, a Constraint Programming (CP) model, and a Maximum Satisfiability (MaxSAT) model. Additionally, we specialized and enhanced a state-of-the-art greedy algorithm. For most network topologies, all our exact models were able to find the optimal placement for both problems. Compared to the current state-of-the-art, an exact approach proved to be valuable, often outperforming the MNMP greedy algorithm by finding better solutions, frequently the optimal ones.

References

1. Bartolini, N., He, T., Arrigoni, V., Massini, A., Trombetti, F., Khamfroush, H.: On Fundamental Bounds on Failure Identifiability by Boolean Network Tomography. IEEE/ACM Transactions on Networking **28**(2), 588–601 (Apr 2020). https://doi.org/10.1109/TNET.2020.2969523
2. Bezerra, P., Chen, P.Y., McCann, J.A., Yu, W.: Adaptive Monitor Placement for Near Real-time Node Failure Localisation in Wireless Sensor Networks. ACM Transactions on Sensor Networks **18**(1), 2:1–2:41 (Oct 2021). https://doi.org/10.1145/3466639, `https://doi.org/10.1145/3466639`
3. CAIDA: The CAIDA Macroscopic Internet Topology Data Kit (Feb 2022), `https://www.caida.org/catalog/datasets/internet-topology-data-kit`
4. Chu, Y., Cai, S., Luo, C.: Nuwls-c-2023: Solver description. MaxSAT Evaluation 2023 p. 23 (2023)
5. Fortz, B., Thorup, M.: Internet traffic engineering by optimizing ospf weights. In: Proceedings IEEE INFOCOM 2000. conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064). vol. 2, pp. 519–528. IEEE (2000)
6. Gay, S., Schaus, P., Vissicchio, S.: Repetita: Repeatable experiments for performance evaluation of traffic-engineering algorithms (2017)

7. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2023), https://www.gurobi.com
8. Hartert, R., Schaus, P., Vissicchio, S., Bonaventure, O.: Solving segment routing problems with hybrid constraint programming techniques. In: Principles and Practice of Constraint Programming: 21st International Conference, CP 2015, Cork, Ireland, August 31–September 4, 2015, Proceedings 21. pp. 592–608. Springer (2015)
9. Hartert, R., Vissicchio, S., Schaus, P., Bonaventure, O., Filsfils, C., Telkamp, T., Francois, P.: A declarative and expressive approach to control forwarding paths in carrier-grade networks. ACM SIGCOMM computer communication review **45**(4), 15–28 (2015)
10. He, T., Bartolini, N., Khamfroush, H., Kim, I., Ma, L., La Porta, T.: Service placement for detecting and localizing failures using end-to-end observations. In: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). pp. 560–569 (2016). https://doi.org/10.1109/ICDCS.2016.21
11. He, T., Ma, L., Swami, A., Towsley, D.: Network Tomography: Identifiability, Measurement Design, and Network State Inference. Cambridge University Press (2021)
12. He, Y., Faloutsos, M., Krishnamurthy, S.: Quantifying routing asymmetry in the internet at the as level. In: IEEE Global Telecommunications Conference, 2004. GLOBECOM '04. vol. 3, pp. 1474–1479 Vol.3 (2004). https://doi.org/10.1109/GLOCOM.2004.1378227
13. Hopcroft, J., Tarjan, R.: Algorithm 447: Efficient algorithms for graph manipulation. Commun. ACM **16**(6), 372–378 (jun 1973). https://doi.org/10.1145/362248.362272, https://doi.org/10.1145/362248.362272
14. John, W., Dusi, M., claffy, k.: Estimating Routing Symmetry on Single Links by Passive Flow Measurements. Tech. rep., ACM International Workshop on TRaffic Analysis and Classification (TRAC) (Mar 2010)
15. Katz, D., Ward, D.: Bidirectional Forwarding Detection (BFD). RFC 5880 (Jun 2010). https://doi.org/10.17487/RFC5880, https://www.rfc-editor.org/info/rfc5880
16. Knight, S., Nguyen, H., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. Selected Areas in Communications, IEEE Journal on **29**(9), 1765 –1775 (october 2011). https://doi.org/10.1109/JSAC.2011.111002
17. Lee, Y., Seok, Y., Choi, Y., Kim, C.: A constrained multipath traffic engineering scheme for mpls networks. In: 2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333). vol. 4, pp. 2431–2436. IEEE (2002)
18. Ma, L., He, T., Leung, K.K., Swami, A., Towsley, D.: Monitor placement for maximal identifiability in network tomography. In: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. pp. 1447–1455 (2014). https://doi.org/10.1109/INFOCOM.2014.6848079
19. Ma, L., He, T., Swami, A., Towsley, D., Leung, K.K.: On optimal monitor placement for localizing node failures via network tomography. Performance Evaluation **91**, 16–37 (Sep 2015). https://doi.org/10.1016/j.peva.2015.06.003, https://www.sciencedirect.com/science/article/pii/S0166531615000516
20. Ogino, N., Kitahara, T., Arakawa, S., Hasegawa, G., Murata, M.: Decentralized boolean network tomography based on network partitioning. In: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium. pp. 162–170 (2016). https://doi.org/10.1109/NOMS.2016.7502809
21. Perron, L., Didier, F.: Cp-sat, https://developers.google.com/optimization/cp/cp_solver/

22. Ren, W., Dong, W.: Robust network tomography: K-Identifiability and monitor assignment. In: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. pp. 1–9 (Apr 2016). https://doi.org/10.1109/INFOCOM.2016.7524375
23. RIPE NCC: RIPE Atlas, `https://atlas.ripe.net`, Accessed: July 2023
24. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with rocketfuel. ACM SIGCOMM Computer Communication Review **32**(4), 133–145 (Aug 2002). https://doi.org/10.1145/964725.633039, `https://doi.org/10.1145/964725.633039`
25. Stanic, S., Subramaniam, S., Sahin, G., Choi, H., Choi, H.A.: Active monitoring and alarm management for fault localization in transparent all-optical networks. IEEE Transactions on Network and Service Management **7**(2), 118–131 (Jun 2010). https://doi.org/10.1109/TNSM.2010.06.I9P0343
26. de Vries, W., Santanna, J.J., Sperotto, A., Pras, A.: How asymmetric is the internet? In: Latré, S., Charalambides, M., François, J., Schmitt, C., Stiller, B. (eds.) Intelligent Mechanisms for Network Configuration and Security. pp. 113–125. Springer International Publishing, Cham (2015)
27. Zhang, R., Newman, S., Ortolani, M., Silvestri, S.: A network tomography approach for traffic monitoring in smart cities. IEEE Transactions on Intelligent Transportation Systems **19**(7), 2268–2278 (2018). https://doi.org/10.1109/TITS.2018.2829086