On the Parameterized Complexity of the Edge Monitoring Problem

Julien Baste[a,*], Fairouz Beggas[b], Hamamache Kheddouci[b], Ignasi Sau[a]

[a] *AlGCo project-team, CNRS, LIRMM, Montpellier, France.*
[b] *University of Lyon, LIRIS UMR5205 CNRS, Claude Bernard Lyon 1 University*
43 Bd du 11 Novembre 1918, F-69622, Villeurbanne, France.

Abstract

In a graph $G = (V, E)$, a vertex $v \in V$ *monitors* an edge $\{u, u'\} \in E$ if $\{v, u\} \in E$ and $\{v, u'\} \in E$. Given an n-vertex graph $G = (V, E)$, in which each edge is contained in at least one triangle, and an integer k, the Edge Monitoring problem consists in finding a set $S \subseteq V$ of size at most k such that each edge of the graph is monitored by at least one element of S. This problem is known to be NP-hard. We prove that it is also $W[2]$-hard when parameterized by k. Using Bidimensionality Theory, we provide an FPT algorithm running in time $2^{\mathcal{O}(\sqrt{k}\cdot\log(\max_{e\in E}\ \omega(e)))}\cdot n$ for the weighted version of Edge Monitoring when the input graph is restricted to be apex-minor-free, in particular, it applies to planar graphs, and where we additionally impose each edge e to be monitored at least $\omega(e)$ times, and the solution to be contained in a set of selected vertices.

Keywords: Edge Monitoring, parameterized complexity, FPT algorithm, apex-minor-free graph, treewidth, dynamic programming, Bidimensionality Theory.

1. Introduction

Sensor networks are increasingly used in the environment and industry thanks notably to the latest developments in the field of wireless sensor networks in the last few years [1]. The need to observe, analyse and control such type of area is essential to many environmental and scientific applications (e.g. measuring pollution levels, detecting earthquake activity, military surveillance, home health care or assisted living...). Anticipating security problems allows to protect the network from a variety of attacks. Many approachs have been proposed to protect sensor networks [2, 3, 4]. In this paper we are interested in the Edge Monitoring mechanism for the security of wireless sensor networks. The basic idea of the Edge Monitoring problem (or watchdog technique) [5, 6, 7] is to select some nodes as monitors in a given sensor network. These monitors are employed for carrying out monitoring operations by listening promiscuously to the transmission of two nodes. They can also perform basic operations of communication and sensing in the network.

The idea is illustrated in Figure 1. Each node in the network has a transmission range. The monitors (or watchdogs) are placed in the intersection of the transmission ranges of the sending (S) and the receiving (R) nodes. They monitor nodes by listening promiscuously to the transmissions of both nodes. When node S forwards a message to R, the watchdog of this link verifies that node R also forwards the message. If R does not forward the message, then it is misbehaving. Similar to this, monitoring nodes are able to detect any malicious actions such as delaying, dropping, modifying, or even fabricated packets.

The Edge Monitoring problem was introduced in sensor networks [7, 8] as self-monitoring. Self-monitoring is an effective mechanism for the security of wireless sensor networks. Dong *et al.* studied

*Corresponding author

Email addresses: baste@lirmm.fr (Julien Baste), fairouz.beggas@liris.cnrs.fr (Fairouz Beggas), hamamache.kheddouci@liris.cnrs.fr (Hamamache Kheddouci), ignasi.sau@lirmm.fr (Ignasi Sau)

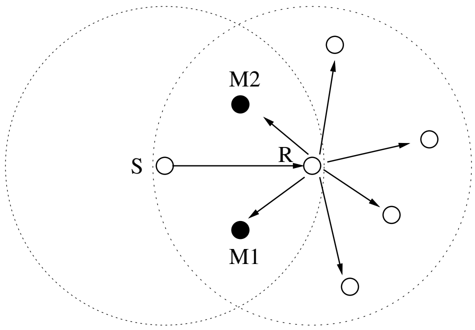

Figure 1: An example to illustrate the EDGE MONITORING problem.

the problem by modeling the communication network as a unit disk graph (*UDG*) [9]. They propose a polynomial-time approximation scheme for the problem in *UDG* graphs with a geometric representation [8].

In [10, 11, 12], the authors concentrated on the system-level fault diagnosis of the network, especially detecting node failures as self-protection. The authors of [7, 8] focused on the fundamental issue of designing an edge self-monitoring topology, where every transmission link can be monitored by nodes within the network. The problem of EDGE MONITORING can be defined from a graph-theoretical point of view as follows. Let $G = (V, E)$ be a graph, with $|V| = n$ and $|E| = m$, and $\omega : E \to \mathbb{N}$ a weight function on the edges. We call $\omega(e)$ the *weight* of the edge e. A node $v \in V$ can *monitor* an edge $e \in E$ if both end-nodes of e are neighbors of v, *i.e.*, e together with v form a triangle in the graph G. An *edge monitoring* of G with weight function ω is a set of vertices such that each edge e of the graph is monitored by at least $\omega(e)$ vertices of the set. The *size* of an edge monitoring is the number of monitors in the set.

In this paper, we study the EDGE MONITORING problem from the perspective of parameterized complexity; see [13, 14, 15]. Parameterized complexity can be seen as a refinement of classical complexity theory in which one takes into account not only the total input size n, but also other aspects of the problem encoded in a parameter k. It is studied as an approach to the exact resolution of NP-complete problems. *Fixed-Parameterized Tractable* (FPT for short) algorithms are used to solve combinatorial optimization problems, including graph algorithms. A problem defined on an n-vertex graph is *fixed-parameter tractable* with respect to a parameter k if it can be solved in FPT-*time*, i.e., in time $f(k) \cdot n^{\mathcal{O}(1)}$, for some computable function f. To the best of our knowledge, the use of parameterized complexity for solving sensor networks problem like EDGE MONITORING has never been done before.

This paper is organized as follows: in Section 2 we introduce some basic definitions and recall the definition of the EDGE MONITORING problem. In Section 3 we prove that the EDGE MONITORING problem is $W[2]$-hard when parameterized by the size of the solution. In Section 4 we present two algorithms that solve a more general problem, namely WEIGHTED EDGE MONITORING. The first one solves the version of the problem parameterized by the treewidth in time $2^{\mathcal{O}(\mathbf{tw}^2 \cdot \log(\max_{e\in E} \omega(e)))} \cdot n$ where $\mathbf{tw}$ is the treewidth of the input graph and $\omega : E \to \mathbb{N}$ is a weight function such that each edge e should be monitored $\omega(e)$ times. The second one solves the version of the problem parameterized by k, the size of the solution, in time $2^{\mathcal{O}(\sqrt{k} \cdot \log(\max_{e\in E} \omega(e)))} \cdot n$ when the input graph is apex-minor-free, in particular, when it is planar, by using Bidimensionality Theory [16, 17, 18]. Section 5 concludes the paper.

2. Notation and preliminaries

In this section we introduce some basic definitions. All the graphs we consider are undirected and contain neither loops nor multiple edges. In this paper, a graph is *triangulated* if any edge is in at least one triangle. We denote by $V(G)$ the set of vertices of a graph G and by $E(G)$ its set of edges. Let $G = (V, E)$ be a graph and $V' \subseteq V$. We denote by $G[V']$ the subgraph of G induced by the vertices V'. We define the *neighborhood* of a vertex v as the set $N(v) = \{c \in V | \{c, v\} \in E\}$. We say that c *monitors* a vertex v if $c \in N(v)$. We define the *neighborhood* of an edge $\{a,b\} \in E$ as the set $N(\{a,b\}) = \{c \in V | \{c,a\} \in E, \{c,b\} \in E\}$. We say that c *monitors* an edge $\{a,b\}$ if $c \in N(\{a,b\})$.

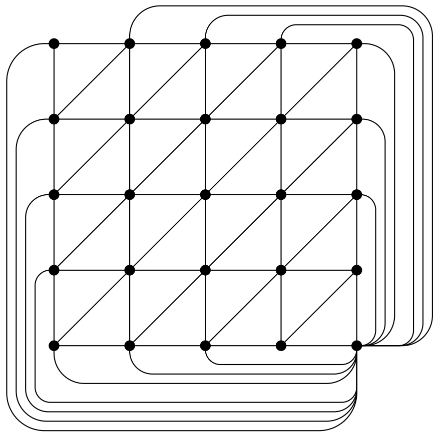

Figure 2: The triangulated grid Γ_5.

Let k be an integer. The *triangulated grid* of size k is the graph $\Gamma_k = (V_k, E_k)$ such that $V_k = \{\ell_{i,j} | 1 \leq i,j \leq k\}$ and $E_k = \{\{\ell_{i,j}, \ell_{i+1,j}\} | 1 \leq i \leq k-1, 1 \leq j \leq k\} \cup \{\{\ell_{i,j}, \ell_{i,j+1}\} | 1 \leq i \leq k, 1 \leq j \leq k-1\} \cup \{\{\ell_{1,j}, \ell_{k,k}\}, \{\ell_{k,j}, \ell_{k,k}\} | 1 \leq j \leq k\} \cup \{\{\ell_{i,1}, \ell_{k,k}\}, \{\ell_{i,k}, \ell_{k,k}\} | 1 \leq i \leq k\}$. Note that Γ_k is triangulated. For an illustration, the graph Γ_5 is depicted in Figure 2. If $i_0, j_0 \in \{1, \ldots, k-1\}$, we call *the square* (i_0, j_0) of Γ_k the set $\{\ell_{i_0,j_0}, \ell_{i_0+1,j_0}, \ell_{i_0,j_0+1}, \ell_{i_0+1,j_0+1}\}$ and the *diagonal* (i_0, j_0) the edge $\{\ell_{i_0+1,j_0}, \ell_{i_0,j_0+1}\}$.

Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be two graphs. We say that H is a *contraction* of G if we can partition V_G into $|V_H|$ sets $(R_u)_{u \in V_H}$ such that for all $u \in V_H$, R_u is not empty and $G[R_u]$ is connected, and such that $\{u_1, u_2\} \in E_H$ if and only if there exist $v_1 \in R_{u_1}$ and $v_2 \in R_{u_2}$ such that $\{v_1, v_2\} \in E_G$.

Treewidth. A *tree-decomposition* of width w of a graph $G = (V, E)$ is a pair $(\mathcal{T}, \sigma)$, where $\mathcal{T}$ is a tree and $\sigma = \{B_t | B_t \subseteq V, t \in V(\mathcal{T})\}$ such that:

- $\bigcup_{t \in V(\mathcal{T})} B_t = V$,
- For every edge $\{u,v\} \in E$ there is a $t \in V(\mathcal{T})$ such that $\{u,v\} \subseteq B_t$,
- $B_i \cap B_k \subseteq B_j$ for all $\{i,j,k\} \subseteq V(\mathcal{T})$ such that j lies on the path $i \ldots k$ in $\mathcal{T}$, and
- $\max_{i \in V(\mathcal{T})} |B_t| = w + 1$.

A tree-decomposition rooted at a node t_r is *nice* if the following conditions are fulfilled:

- $B_{t_r} = \emptyset$,
- each node has at most two children,
- for each leaf $t \in V(\mathcal{T})$, $B_t = \emptyset$,
- if $t \in V(\mathcal{T})$ has exactly one child t', then either
  - $B_t = B_{t'} \cup \{v\}$ for some $v \notin B_{t'}$ and this node is called an *introduce vertex*, or
  - $B_t = B_{t'} \setminus \{v\}$ for some $v \in B_{t'}$ and this node is called a *forget vertex*, and
- if $t \in V(\mathcal{T})$ has exactly two children t' and t'', then $B_t = B_{t'} = B_{t''}$. This node is called a *join vertex*.

The sets B_t are called *bags*. The *treewidth* of G, denoted by $\mathbf{tw}(G)$, is the smallest integer w such that there is a tree-decomposition of G of width w. When context is clear we will use the notation $\mathbf{tw}$ instead of $\mathbf{tw}(G)$. An *optimal tree-decomposition* is a tree-decomposition of width $\mathbf{tw}(G)$. Moreover, if we have a tree-decomposition, then we can build a nice tree-decomposition of G with the same width in polynomial time [19].

In the paper we are interested in the following problem:

EDGE MONITORING **Input:** A triangulated graph $G = (V, E)$ and an integer k. **Output:** A set $S \subseteq V$ of size at most k such that $\forall e \in E$, $S \cap N(e) \neq \emptyset$.

Note that we restrict EDGE MONITORING to apply only on triangulated graph. Indeed if the graph is not triangulated, then we can directly answer that the problem has no solution. This restriction is no big deal because if the graph is not triangulated then, in practice, either we add sensors that cover the edges that are not in a triangle or remove the edges by forbidden the communication by these edges.

3. $W[2]$-hardness of EDGE MONITORING when parameterized by k

In this section we show that the problem is $W[2]$-hard when parameterized by the size of the solution. In order to prove that, we reduce from RED-BLUE DOMINATING SET, which is known to be $W[2]$-hard [13].

RED-BLUE DOMINATING SET **Input:** A graph $G = (V, E)$, a partition (V_r, V_b) of V, and an integer k. **Output:** A set $S \subseteq V_b$ of size at most k such that $\forall v \in V_r$, $S \cap N(v) \neq \emptyset$.

Theorem 1. EDGE MONITORING *is $W[2]$-hard parameterized by the size of the solution.*

Proof: Let $G = (V, E)$ be a graph, let (V_r, V_b) be a partition of V, and let k be an integer. We want to solve RED-BLUE DOMINATING SET on (G, V_r, V_b, k). Without lost of generality, we can assume that there is no isolated vertex.

We construct from (G, V_r, V_b, k) the graph $G' = (V', E')$ as depicted in Figure 3. Formally, $V' = V'_b \cup V'_e \cup V_a$ where $V'_b = \{v^b | v \in V_b\}$, $V'_e = \{v^1 | v \in V_r\} \cup \{v^2 | v \in V_r\}$, $V_a = \{a^i_j, b^i_j, c^i_j | i \in \{1,2\}, j \in \{1,2,3\}\}$, and $E' = \{\{v^1, v^2\} | v \in V_r\} \cup \{\{v^b, w^1\} | \{v, w\} \in E\} \cup \{\{v^b, w^2\} | \{v, w\} \in E\} \cup \{\{a^i_j, v^i\} | i \in \{1,2\}, j \in \{1,2,3\}\} \cup \{\{a^i_j, v^b\} | i \in \{1,2\}, j \in \{1,2,3\}\} \cup \{\{a^i_j, a^i_{j'}\} | i \in \{1,2\}, j, j' \in \{1,2,3\}, j \neq j'\} \cup \{\{a^i_j, b^i_j\}, \{a^i_j, c^i_j\}, \{b^i_j, c^i_j\} | i \in \{1,2\}, j \in \{1,2,3\}\}$.

We now show that solving RED-BLUE DOMINATING SET on (G, V_r, V_b, k) is equivalent to solving EDGE MONITORING on $(G', k + 18)$. Let S be a solution of RED-BLUE DOMINATING SET on (G, V_r, V_b, k). Let $S' = \{v^b | v \in S\} \cup V_a$. Then S' is a solution of EDGE MONITORING on $(G', k + 18)$. Indeed, $|S'| \leq k + 18$ by definition of S and V_a. Let $e \in E'$. If $e = \{v^1, v^2\}$ with $v \in V_r$, then by definition of S, there exists $t \in S$ that is neighbor of v in G, so t^b monitors e in G'. If $e = \{v^b, w^1\}$ with $v \in V_b$ and $w \in V_r$, then a^1_1 monitors e. The same happens if $e = \{v^b, w^2\}$. If $e = \{a^i_j, v^i\}$ then $a^i_{(j \mod 3)+1}$ monitors e. As $\{a^i_1, a^i_2, a^i_3\}$ is a triangle where all the vertices are in S', all the edges are monitored. The same happens for the triangles $\{a^i_j, b^i_j, c^i_j\}$, $i \in \{1,2\}$, $j \in \{1,2,3\}$.

Now let S' be a solution of EDGE MONITORING on $(G', k + 18)$. For each $i \in \{1,2\}$ and $j \in \{1,2,3\}$, the edges $\{a^i_j, b^i_j\}$, $\{a^i_j, c^i_j\}$, and $\{b^i_j, c^i_j\}$ can be monitored only by the vertices c^i_j, b^i_j, and a^i_j respectively. So they need to be in S'. One can check that the only edges not monitored by V_a are the edges of the form $\{v^1, v^2\}$, and by construction of G' the only vertices that can monitor them are vertices from V'_b. It directly follows that $S = \{v \in V_b | v^b \in S'\}$ is a solution of RED-BLUE DOMINATING SET on (G, V_r, V_b, k). □

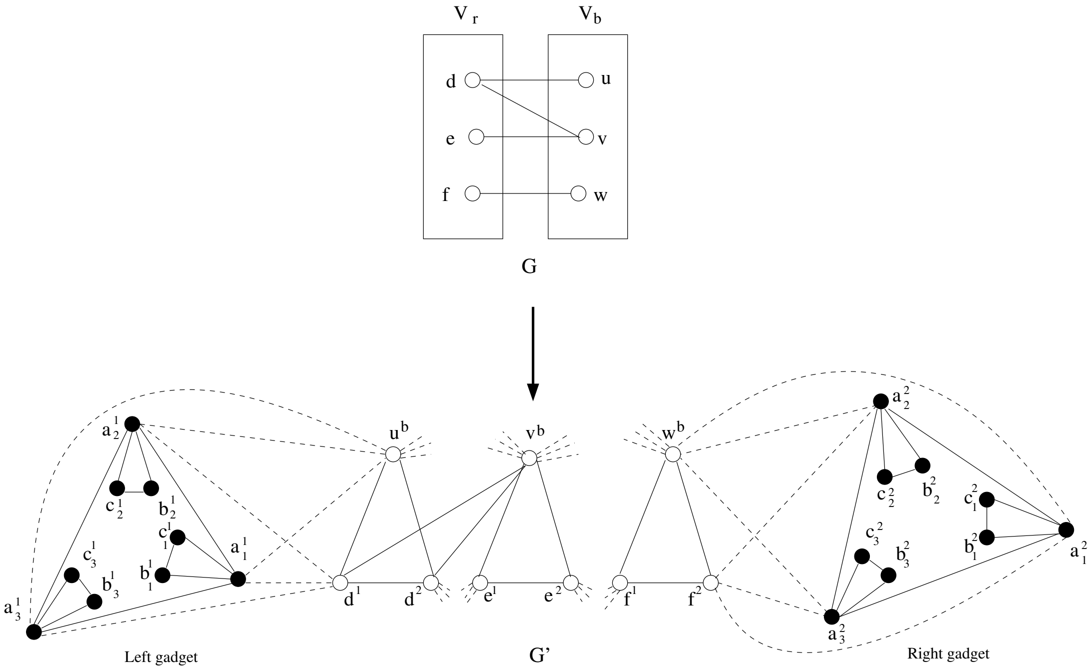

Figure 3: Edge Monitoring gadget. In the figure, the vertices u^b, v^b, w^b, d^1, e^1, and f^1 are connected to the three vertices a_1^1, a_2^1, and a_3^1 like u^b and d^1 are, and the vertices u^b, v^b, w^b, d^2, e^2, and f^2 are connected to the three vertices a_1^2, a_2^2, and a_3^2 like w^b and e^2 are.

4. Fixed-parameter algorithms for Edge Monitoring

In the following, we will present algorithms that solve the Edge Monitoring problem. The first one is parameterized by the treewidth of the input graph and the second one, based on the first one, uses Bidimensionality to solve Edge Monitoring parameterized by the size of the solution when the input graph is apex-minor-free. In order to be as general as possible, we will solve a more general problem, namely Weighted Edge Monitoring.

> Weighted Edge Monitoring
> **Input:** A triangulated graph $G = (V, E)$, an integer k, a set $M \subseteq V$, and a weight function $\omega : E \to \{1, \ldots, k\}$.
> **Output:** A set $S \subseteq M$ of size at most k such that $\forall e \in E$, $|S \cap N(e)| \geq \omega(e)$.

In this version, we allow only some selected monitors to be in the solution, and we impose that each edge is monitored by at least a given number of monitors.

From Theorem 1, we directly obtain the following.

Corollary 1. Weighted Edge Monitoring *is $W[2]$-hard parameterized by k.*

We now focus on the algorithms. First we present an FPT algorithm parameterized by the treewidth.

Lemma 1. *Let $G = (V, E)$ be a graph, k be an integer, M be a subset of V, and $\omega : E \to \{1, \ldots, k\}$ be a weight function.* Weighted Edge Monitoring *on (G, k, M, ω) can be solved in time $2^{\mathcal{O}(\mathbf{tw}^2 \cdot \log(\max_{e \in E} \omega(e)))} \cdot n$, where $\mathbf{tw}$ is the treewidth of G.*

Proof: Let $G = (V, E)$ be a triangulated graph, k be an integer, M be a subset of V, $\omega : E \to \{1, \ldots, k\}$ be a weight function, and $(\mathcal{T}, \mu)$ be a nice tree-decomposition of G rooted at a node t_r of width $\mathbf{tw}$.

For each $t \in V(\mathcal{T})$, we denote by V_t the set of vertices of all descendents of t, $G_t = G[V_t]$, and $E_t = E(G[B_t])$. Note that this graph may be disconnected.

We use a dynamic programming approach. The table we store at a node t will contain elements of the form (X, Y, p), where $X \subseteq B_t$ is the set of chosen vertices in B_t for this solution, $Y \subseteq E_t \times \mathbb{N}$ is the set of pairs (y, m) where the edge y still needs to be monitored m times in G_t, and p is the number of vertices we already have chosen. We will keep such an element in the table, if there exists a solution S of our problem of size at most k such that $S \cap B_t = X$, $|S \cap V_t| \leq p$, $S \cap V_t$ monitors all the edges of $E(G_t) \setminus \{y | \exists m \in \mathbb{N} : (y, m) \in Y\}$, and for each $(y, m) \in Y$, $S \cap V_t$ monitors $\omega(y) - m$ times the edge y. Formally, if $H = (V_h, E_h)$ is a graph, $B \subseteq V_h$, $X \subseteq B$, and $Y \subseteq E(H[B]) \times \{1, \dots, k\}$, we define $sol(H, B, X, Y, p, M) = true$, if and only if there exists a set $S \subseteq V_h \cap M$ of size at most p such that for each $(e, m) \in Y$, $|S \cap N(e)| = \omega(e) - m$, and for each $e \in E_h \setminus \{y | \exists m \in \mathbb{N} : (y, m) \in Y\}$, $|S \cap N(e)| = \omega(e)$, and $S \cap B = X$. We define the table we store at each node $t \in V(\mathcal{T})$ to be $\mathcal{R}_t = \{(X, Y, p) | X \subseteq B_t, Y \subseteq E(G[B_t]) \times \{1, \dots, k\}, sol(G_t, B_t, X, Y, p, M), p \leq k\}$. Note that there is a solution of our problem if and only if $\mathcal{R}_{t_r} \neq \emptyset$. For convenience, if $(X, Y, p) \in \mathcal{R}_t$ and $(X, Y, q) \in \mathcal{R}_t$ with $p < q$ then our algorithm will keep only (X, Y, p), as the other entry is not relevant. Let $t \in V(\mathcal{T})$. We can compute $\mathcal{R}_t$ as follows:

- If t is a leaf then $G_t = (\emptyset, \emptyset)$ and $\mathcal{R}_t = \{(\emptyset, \emptyset, 0)\}$.
- If t is an introduce vertex v and $v \in M$, let t' be its child. Then $\mathcal{R}_t = \{(X \cup \{v\}, \{(y, m - |N(y) \cap \{v\}|) | (y, m) \in Y\} \cup \{(\{v, w\}, m') | w \in B_t, \{v, w\} \in E, m' = \max(\omega(\{v, w\}) - |N'\{v, w\} \cap X|, 0)\}, p + 1) | (X, Y, p) \in \mathcal{R}_{t'}, p + 1 \leq k\} \cup \{(X, Y \cup \{(\{v, w\}, m') | w \in B_t, \{v, w\} \in E, m' = \max(\omega(\{v, w\}) - |N'\{v, w\} \cap X|, 0)\}, p) | (X, Y, p) \in \mathcal{R}_{t'}\}$.
- If t is an introduce vertex v and $v \notin M$, let t' be its child. Then $\mathcal{R}_t = \{(X, Y \cup \{(\{v, w\}, m') | w \in B_t, \{v, w\} \in E, m' = \max(\omega(\{v, w\}) - |N'\{v, w\} \cap X|, 0)\}, p) | (X, Y, p) \in \mathcal{R}_{t'}\}$.
- If t is a forget vertex v, let t' be its child. Then $\mathcal{R}_t = \{(X \setminus \{v\}, Y \setminus \{(\{v, w\}, 0) | w \in B_t, (\{v, w\}, 0) \in Y\}, p) | (X, Y, p) \in \mathcal{R}_{t'}, \forall w \in X, m \in \{1, \dots, k\} : (\{v, w\}, m) \notin Y\}$. Note that if $v \notin X$ then $X \setminus \{v\} = X$.
- If t is a join vertex, let t' and t'' be its children. Then $\mathcal{R}_t = \{(X' \cup X'', \{(y, m) | (y, m') \in Y', (y, m'') \in Y'', m = m' + m'' - \omega(y) + |N(y) \cap (X' \cap X'')|\}, p' + p'' - |X' \cap X''|) | (X', Y', p') \in \mathcal{R}_{t'}, (X'', Y'', p'') \in \mathcal{R}_{t''}, p' + p'' - |X' \cap X''| \leq k\}$.

For all $t \in V(\mathcal{T})$, if $(X, Y, p) \in \mathcal{R}_t$ then $X \subseteq B_t$ and $Y \subseteq E_t \times \{1, \dots, \max_{e \in E} \omega(e)\}$. Note that if (y, m) and (y, m') are in Y with $m < m'$, then we need to keep only (y, m). So we can see Y as a subset of all functions $E_t \to \{1, \dots, \max_{e \in E} \omega(e)\}$. We obtain that $|Y| \leq 2^{\mathbf{tw}^2 \cdot \log(\max_{e \in E} \omega(e))}$. Thus, $|\mathcal{R}_t| \leq 2^{\mathbf{tw}} \cdot 2^{\mathbf{tw}^2 \cdot \log(\max_{e \in E} \omega(e))}$. So we can solve Edge Monitoring on (G, k) in time $2^{\mathcal{O}(\mathbf{tw}^2 \cdot \log(\max_{e \in E} \omega(e)))} \cdot n$. ◻

If G is apex-minor-free, then, there exists a constant a, depending only on the apex-graph, such that $|E| \leq a|V|$ [20]. In particular, it implies that in the previous complexity analysis, if G is apex-minor-free, then Y is of size at most $a|V| \cdot \log(\max_{e \in E} \omega(e))$. This directly gives the following lemma.

Lemma 2. *Let $G = (V, E)$ be a apex-minor-free graph, k be an integer, M be a subset of V, and $\omega : E \to \{1, \dots, k\}$ be a weight function.* Weighted Edge Monitoring *on (G, k, M, ω) can be solved in time $2^{\mathcal{O}(\mathbf{tw} \cdot \log(\max_{e \in E} \omega(e)))} \cdot n$.*

Theorem 2 ([21]). *There exists a constant c such that for every apex-minor-free graph G and every integer k such that $k \leq \frac{\mathbf{tw}(G)}{c}$, the triangulated grid Γ_k is a contraction of G.*

Theorem 3. *Let $G = (V, E)$ be a apex-minor-free graph, k be an integer, ω be a weight function $\omega : E \to \{1, \dots, k\}$, and M be a subset of V.* Weighted Edge Monitoring *on (G, k), can be solved in time $2^{\mathcal{O}(\sqrt{k} \cdot \log(\max_{e \in E} \omega(e)))} \cdot n$.*

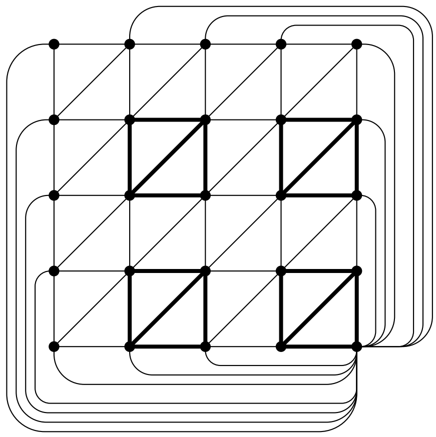

Figure 4: The considered squares in Γ_5 and their diagonals.

Proof: Let $G = (V, E)$ be a apex-minor-free graph and k be an integer. Assume first that $\mathbf{tw} > c(2\lceil\sqrt{(k+1)}\rceil + 2)$. By Theorem 2, $\Gamma_{(2\lceil\sqrt{(k+1)}\rceil+2)}$ is a contraction of G. Let $L = \{\ell_{i,j} | i, j \in \mathbb{N}, 1 \leq i, j \leq (2\lceil\sqrt{(k+1)}\rceil + 2)\}$ be the vertex set of $\Gamma_{(2\lceil\sqrt{(k+1)}\rceil+2)}$, and let M be its edge set. Let $(R_u)_{u\in L}$ be a partition of V such that for all $u \in L$, R_u is not empty, $G[R_u]$ is connected, and such that $\{u_1, u_2\} \in E(\Gamma_{(2\lceil\sqrt{(k+1)}\rceil+2)})$ if and only if there exist $v_1 \in R_{u_1}$ and $v_2 \in R_{u_2}$ such that $\{v_1, v_2\} \in E$.

Consider the $\lceil\sqrt{k+1}\rceil^2$ squares $(2i, 2j)$, for $1 \leq i \leq \lceil\sqrt{k+1}\rceil$ and $1 \leq j \leq \lceil\sqrt{k+1}\rceil$. For simplicity we denote by $Q_{i,j}$ the square $(2i, 2j)$. The selected squares are illustrated in Figure 4. By construction, the squares $Q_{i,j}$ are pairwise vertex-disjoint. For each i, j, we arbitrarily choose $e_{i,j} = \{a_{i,j}, b_{i,j}\} \in E$ such that $a_{i,j} \in R_{2i+1,2j}$ and $b_{i,j} \in R_{2i,2j+1}$. We denote by $e_{i,j}$ the representative edge of $Q_{i,j}$. The edge $e_{i,j}$ can be monitored only by an element of $R_{\ell_{2i,2j}} \cup R_{\ell_{2i,2j+1}} \cup R_{\ell_{2i+1,2j}} \cup R_{\ell_{2i+1,2j+1}}$, because the other $\ell_{i',j'}$ are not connected to both $\ell_{2i+1,2j}$ and $\ell_{2i,2j+1}$. Thus, there are no two distinct representative edges in G that can be monitored by the same vertex of G. This means that the solution should be of size at least $k + 1$, that is the number of squares we had consider. As we ask for a solution of size at most k, then we can safely answer that there is no such a solution.

Now assume that $\mathbf{tw}(G) \leq c(2\lceil\sqrt{(k+1)}\rceil + 2)$. By Lemma 2, we know that there is an algorithm in time $2^{\mathcal{O}(\mathbf{tw})\cdot\log(\max_{e\in E}\omega(e)))} \cdot n$ to solve the problem. In particular, this algorithm runs in time $2^{\mathcal{O}(\sqrt{k}\cdot\log(\max_{e\in E}\omega(e)))} \cdot n$.
□

5. Conclusion and further research

In this paper we studied the Edge Monitoring problem under the approach of parameterized complexity. We showed that, in general graphs, we are unlikely to be able to solve our problem in FPT time when parameterized by the size of the solution. We used Bidimensionality to show that if the input graph has the topological restriction to be apex-minor-free, then our problem can be solved in time $2^{\mathcal{O}(\sqrt{k})} \cdot n$. We even show that the weighted version of the problem, Weighted Edge Monitoring, can be solved in a similar time, i.e., in time $2^{\mathcal{O}(\sqrt{k}\cdot\log(\max_{e\in E}\omega(e)))} \cdot n$, when the input graph is apex-minor-free. A natural extension is to consider H-minor-free graphs for a general graph H, not necessarily an apex graph, and even the larger classes of H-topological-minor-free graphs".

Sensor networks can be modeled by many classes of graphs. Some of them can be modeled by planar graphs, that are also apex-minor-free, but there are other interesting classes of graphs that correspond to real sensor networks. For instance, unit disk graphs constitute a model for wireless sensor networks [9, 22]. It is currently not known whether FPT algorithms exist for this class of graphs.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer networks 38 (4) (2002) 393–422.

[2] H.-S. Lim, G. Ghinita, E. Bertino, M. Kantarcioglu, A game-theoretic approach for high-assurance of data trustworthiness in sensor networks, in: Data Engineering (ICDE), 2012 IEEE 28th International Conference on, IEEE, 2012, pp. 1192–1203.

[3] M. Rezvani, A. Ignjatovic, E. Bertino, S. Jha, Secure data aggregation technique for wireless sensor networks in the presence of collusion attacks, Dependable and Secure Computing, IEEE Transactions on 12 (1) (2015) 98–110.

[4] X. Hong, P. Wang, J. Kong, Q. Zheng, J. Liu, Effective probabilistic approach protecting sensor traffic, in: Military Communications Conference, 2005. MILCOM 2005. IEEE, IEEE, 2005, pp. 169–175.

[5] B. Neggazi, M. Haddad, V. Turau, H. Kheddouci, A self-stabilizing algorithm for edge monitoring problem, in: Stabilization, Safety, and Security of Distributed Systems, Springer, 2014, pp. 93–105.

[6] G. Wei, Z. Zhu, Y. Mao, N. Xiong, A distributed node self-monitoring mechanism in wireless sensor networks, in: Information Science and Engineering (ICISE), 2010 2nd International Conference on, IEEE, 2010, pp. 1684–1687.

[7] D. Dong, X. Liao, Y. Liu, C. Shen, X. Wang, Edge self-monitoring for wireless sensor networks, Parallel and Distributed Systems, IEEE Transactions on 22 (3) (2011) 514–527.

[8] D. Dong, Y. Liu, X. Liao, Self-monitoring for sensor networks, in: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2008, pp. 431–440.

[9] B. N. Clark, C. J. Colbourn, D. S. Johnson, Unit disk graphs, Discrete mathematics 86 (1) (1990) 165–177.

[10] C. Hsin, M. Liu, Self-monitoring of wireless sensor networks, Computer Communications 29 (4) (2006) 462–476.

[11] D. Wang, Q. Zhang, J. Liu, The self-protection problem in wireless sensor networks, ACM Transactions on Sensor Networks (TOSN) 3 (4) (2007) 20.

[12] Y. Wang, X.-Y. Li, Q. Zhang, Efficient algorithms for p-self-protection problem in static wireless sensor networks, Parallel and Distributed Systems, IEEE Transactions on 19 (10) (2008) 1426–1438.

[13] R. G. Downey, M. R. Fellows, Fundamentals of Parameterized Complexity, Texts in Computer Science, Springer, 2013.

[14] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer-Verlag, 2006.

[15] R. Niedermeier, Invitation to Fixed-Parameter Algorithms, Oxford University Press, 2006.

[16] E. D. Demaine, M. Hajiaghayi, Bidimensionality, in: M. Kao (Ed.), Encyclopedia of Algorithms, Springer, 2008.

[17] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, D. M. Thilikos, Subexponential parameterized algorithms on graphs of bounded-genus and H-minor-free graphs, in: In Proc. 15th SODA, Society for Industrial and Applied Mathematics, 2004, pp. 830–839.

[18] E. Demaine, M. Hajiaghayi, D. Thilikos, The bidimensional theory of bounded-genus graphs, in: J. Fiala, V. Koubek, J. Kratochvíl (Eds.), Mathematical Foundations of Computer Science 2004, Vol. 3153 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 191–203.

[19] T. Kloks, Treewidth, Computations and Approximations, Vol. 842 of Lecture Notes in Computer Science, Springer, 1994.

[20] A. Thomason, The extremal function for complete minors, Journal of Combinatorial Theory, Series B 81 (2) (2001) 318–338.

[21] F. V. Fomin, P. Golovach, D. M. Thilikos, Contraction obstructions for treewidth, Journal of Combinatorial Theory, Series B 101 (5) (2011) 302 – 314.

[22] P. Santi, Topology control in wireless ad hoc and sensor networks, ACM computing surveys (CSUR) 37 (2) (2005) 164–194.