



HAL
open science

Complexity of edge monitoring on some graph classes

Guillaume Bagan, Fairouz Beggas, Mohammed Haddad, Hamamache
Kheddouci

► **To cite this version:**

Guillaume Bagan, Fairouz Beggas, Mohammed Haddad, Hamamache Kheddouci. Complexity of edge monitoring on some graph classes. 2024. hal-02167603

HAL Id: hal-02167603

<https://hal.science/hal-02167603v1>

Preprint submitted on 5 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Complexity of Edge Monitoring on Some Graph Classes

☆

Guillaume Bagan^{a,*}, Fairouz Beggas^a, Mohammed Haddad^a, Hamamache Kheddouci^a

^aUniversité Lyon 1, LIRIS UMR CNRS 5205, F-69621, Lyon, France

Abstract

In this paper, we study the complexity of the edge monitoring problem. A vertex v monitors an edge e if both extremities together with v form a triangle in the graph. Given a graph $G = (V, E)$ and a weight function on edges c where $c(e)$ is the number of monitors that needs the edge e , the problem is to seek a minimum subset of monitors S such that every edge e in the graph is monitored by at least $c(e)$ vertices in S . In this paper, we study the edge monitoring problem on several graph classes such as complete graphs, block graphs, cographs, split graphs, interval graphs and planar graphs. We also generalize the problem by adding weights on vertices.

Keywords: Edge monitoring, weighted edge monitoring, domination, complexity, algorithms, parameterized, approximation.

1. Introduction

In this paper, we are interested in a variant of the dominating set problem: the edge monitoring problem. The edge monitoring (or watchdog technique) is a mechanism for the security of wireless sensor networks [17, 20, 7]. The basic idea is to select some nodes as monitors in a given sensor network. These monitors are employed for carrying out monitoring operations by promiscuously listening to the transmission of two nodes. They can also perform basic operations of communication and sensing in the network.

The edge monitoring problem is defined as follows. Let $G = (V, E)$ be a graph and c be an integer weight function on edges of G . An edge monitoring set of (G, c) is a set of vertices S such that each edge e of G is monitored by at least $c(e)$ vertices of the set S . A node $v \in V$ monitors an edge $e \in E$ if its both end-nodes are neighbors of v *i.e.*, e together with v form a triangle in the graph. Consider the example in Figure 1. The black nodes can monitor all edges depicted in bold.

☆ A part of this paper has been presented at Discrete Mathematics Days 2016 [1]

*Corresponding Author, guillaume.bagan@liris.cnrs.fr

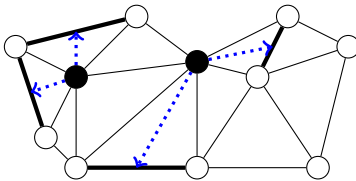


Figure 1: Edge monitoring set of a graph

Dong et al. [7] proved that the edge monitoring problem is NP-complete even restricted to unit disk graphs and they propose a polynomial-time approximation scheme for this class of graphs. Baste et al. [3] focused on parametrized complexity. They proved that the problem is $W[2]$ -hard on general graphs and proposed an FPT algorithm for planar graphs and, more generally, for apex-minor-free graphs.

This paper focuses on the complexity of the edge monitoring problem and its weighted version on different classes of graphs. A weighted version of the edge monitoring problem is applied on graphs with weights on vertices (in addition to weights on edges). Let $(G = (V, E), c, w)$ be a weighted graph with $w(v)$ the weight associated to a vertex $v \in V$. The aim is to find a set S that monitors (G, c) and minimizes $w(S)$.

Among the classes studied in this paper, we consider block graphs, split graphs, cographs and interval graphs which are perfect graphs. Note that the class of complete graphs is included in all graph classes mentioned before. Since we prove that the edge monitoring problem is hard for complete graphs, we consider the problem in these classes with more restricted conditions. We also have a special interest in the unit disc graphs and planar graphs.

This paper is organized as follows. Section 2 gives formal definitions of the problem and its variant. Some basic graph terminologies and concept of complexity are also presented. Section 3 presents some introductory results. In Section 4, we study the problem in complete graphs and block graphs. We give a polynomial time approximation scheme for weighted complete graphs. Sections 5,6,7 are dedicated to interval graphs, cographs and split graphs respectively. In section 8, we prove that the problem is NP-complete on planar unit disk graphs. Besides, we show that there exists a PTAS for Weighted Edge Monitoring on weighted planar graphs and more generally on weighted apex-minor-free families of graphs. The last section summarizes all results of this paper and give some suggestions for further research.

2. Preliminaries

In this section, we give some basic graph terminology and complexity used throughout this paper. We also give definitions of the edge monitoring problem and all concepts used around this problem.

2.1. Basic notions of graphs

Graphs considered in this paper are simple, undirected and without loops. Let $G = (V, E)$ be a graph. The (open) neighborhood of a vertex v is $N(v) = \{u : \{u, v\} \in E\}$. The closed neighborhood of v is $N[v] = N(v) \cup \{v\}$. For a set $S \subseteq V$, $N[S] = \bigcup_{v \in S} N[v]$. The induced graph of G by S , denoted by $G[S] = (S, E')$ contains all the edges of E whose extremities belong to S . A clique is a set $K \subseteq V$ such that each two vertices of K are adjacent. An independent set is a set $S \subseteq V$ such that no edge of G has its two end vertices in S . The clique number of G , denoted by $\omega(G)$, is the cardinality of a maximum clique in G . A graph is chordal if it has no induced cycle of length more than 3. The treewidth of G , denoted by $\text{tw}(G)$, is $\min\{\omega(H) : H \text{ is chordal } \wedge G \text{ is a subgraph of } H\} - 1$. A set $S \subseteq V$ is a dominating set of G if $N[S] = G$. A set $S \subseteq V$ is a total dominating set of G if $N(S) = G$. A set $S \subseteq V$ is a double dominating set of G if for every vertex $x \in V$, $|N[x] \cap S| \geq 2$. $\gamma(G)$ (resp. $\gamma_t(G)$, $\gamma_{\times 2}(G)$) denotes the size of a smallest dominating set (resp. total dominating set, double dominating set) of G or $+\infty$ if such a set does not exist.

2.2. Edge monitoring

Let $e = \{v_1, v_2\}$ be an edge of a graph G . We denote by $M(e)$ the set of vertices v such that $\{v_1, v_2, v\}$ forms a triangle. We say that v monitors e . Let $\alpha \geq 0$ be an integer. A set $S \subseteq V$ α -monitors an edge e if $|M(e) \cap S| \geq \alpha$. Let $G = (V, E)$ be a graph and $c : E \rightarrow \mathbb{N}$ be a weight function over the edges of G . S monitors (G, c) if S $c(e)$ -monitors every edge e in G . The couple (G, c) is called a weighted graph. $\gamma_m(G, c) = \{|S| : S \text{ is a monitoring set of } (G, c)\}$ (and $+\infty$ if no monitoring set exists). $\gamma_m(G) = \gamma_m(G, c)$ where c is 1-uniform.

We define the problem EDGEMONITORING as a decision problem. However, we use the same name for the minimization problem and the parameterized version with k as parameter.

EDGEMONITORING
Input: A weighted graph (G, c) , an integer $k \geq 0$
Question: Is there a monitoring set S of G such that $|S| \leq k$?

Let (G, w, c) such that $G = (V, E)$ is a graph, $w : V \rightarrow \mathbb{Q}^+$ and $c : E \rightarrow \mathbb{N}$. $\gamma_m(G, w, c) = \min\{w(S) : S \text{ monitors } (G, c)\}$. (G, w, c) is also called a weighted graph. Similarly to EDGEMONITORING, we define the problem WEM.

WEM
Input: A weighted graph (G, w, c) , a number $k \in \mathbb{Q}^+$
Question: Is there a monitoring set S of G, c such that $w(S) \leq k$?

Let (G, c) be a weighted graph with $G = (V, E)$. Then $C(G, c) = \max\{c(e) : e \in E\}$. Whenever G and c are obvious from the context, we write C instead of $C(G, c)$. A family of weighted graphs \mathcal{F} is C -bounded if there exists an integer m such that $C(G, c) \leq m$ for every $(G, c) \in \mathcal{F}$.

2.3. Complexity

Let X be a minimization problem. Let $\rho > 1$. An algorithm A is called a ρ -approximation algorithm for X , if, for all instances I of X , it delivers a feasible solution with objective value $A(I)$ such that $A(I) \leq \rho \cdot \text{OPT}(I)$. A polynomial time approximation scheme (PTAS for short) for X is a family of $(1 + \epsilon)$ -approximation algorithms computable in polynomial time in the input size for any $\epsilon > 0$.

Parameterized complexity consists in studying the complexity of problems according to their input size, but also to another parameter. For any basic notions of parameterized complexity ($W[1]$, FPT-reduction, etc.); see [9].

In the following, we prove that 1-uniform EDGEMONITORING cannot be approximated with a constant ratio. We use a reduction from this problem.

TOTALDOMINATINGSET
Input: A graph $G = (V, E)$ without isolated vertex
Output: a minimum total dominating set of G

Theorem 1. *1-uniform EDGEMONITORING cannot be approximated within $(1 - \epsilon) \ln |V|$ for any $\epsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$.*

Proof. It has been proved in [5] that TOTALDOMINATINGSET cannot be approximated within $(1 - \epsilon) \ln |V|$ for any $\epsilon > 0$, unless $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log n)})$. We will define an approximation preserving reduction from TOTALDOMINATINGSET to 1-uniform EDGEMONITORING. Let $G = (V, E)$ be a graph without isolated vertex. We construct G' from G by adding three vertices u, v, w which form a clique and connecting u to every vertex in V . We will prove that $\gamma_m(G') = \gamma_t(G) + 3$.

Let S be a total dominating set of G and $S' = S \cup \{u, v, w\}$. Then S' is a monitoring set of G' . Indeed, the edges uv, uw and vw are monitored by w, v and u respectively. The edges in E are monitored by u . Let x be a vertex in V then x has a neighbor y in S . Thus, ux is monitored by y .

Now, let S be a monitoring set of G' . $\{u, v, w\} \subseteq S$. Otherwise, uv, vw or uw is not monitored by S . Let $S' = S \setminus \{u, v, w\}$. We will prove that S' is a total dominating set of G . Let x be a vertex of G . The edge xu is monitored by a vertex y in S' . Since $\{x, y, u\}$ forms a triangle, x is adjacent to a vertex in S' . Hence, $\gamma_m(G') = \gamma_t(G) + 3$.

Using the same method as in Theorem 1 of [14] we obtain the desired result. \square

3. Complete graphs and block graphs

In this section we present some results of WEM problem on complete graphs and block graphs.

A block graph is a graph where each biconnected component (block) is a clique. The block-cut tree T of a connected graph G is defined as follows. The vertices of T are the blocks and the articulation points of G . There is an edge between an articulation point v and a block B in T if $v \in B$.

Lemma 2. *Let (G, c) be a weighted graph such that $G = (V, E)$ is a complete graph, $C = \max\{c(e) : e \in E\}$ and $|V| \geq C + 2$. Then, $C \leq \gamma_m(G, c) \leq C + 2$. Moreover, every set $S \subseteq V$ with $|S| \geq C + 2$ is a monitoring set of (G, c) .*

Proof. Since there exists an edge e of weight $c(e) = C$, we need C vertices to monitor it. Thus, $C \leq \gamma_m(G)$. Let $S \subseteq V$ be a set such that $|S| \geq C + 2$. Then, every edge e is $c(e)$ -monitored by S . Indeed, let $e = \{u, v\} \in E$. Then, the set $S \setminus \{u, v\}$ of size at least $C \geq c(e)$ $c(e)$ -monitors e . \square

Lemma 3. *Let (G, c) be a weighted graph such that $G = (V, E)$ is a complete graph and c is k -uniform with $k > 0$ and $|V| \geq k + 2$. Then, $\gamma_m(G, c) = k + 2$.*

Proof. Assume, for the sake of contradiction, that there exists a set S that monitors G such that $|S| < k + 2$. If $|S| = 1$, let v be the unique element of S . Let e an edge incident to v . Then, e is not $c(e)$ -monitored by S . Otherwise, let u and v be two elements in S . Then, $M(\{u, v\}) \cap S = |S| - 2 < k$ so $\{u, v\}$ is not monitored by S . \square

Theorem 4. *EDGEMONITORING is NP-complete on complete graphs. Moreover, EDGEMONITORING is $W[1]$ -complete on complete graphs.*

Proof. We will prove that EDGEMONITORING is equivalent to INDEPENDENTSET under FPT-reductions. Since INDEPENDENTSET is $W[1]$ -complete, the results follow.

First, we show a reduction from INDEPENDENTSET to EDGEMONITORING. Let $(G = (V, E), k)$ be an instance of INDEPENDENTSET. Without loss of generality, we can assume that G is connected. Indeed, it is easily seen that INDEPENDENTSET remains $W[1]$ -hard under this restriction. We build an instance $(G' = (V, E'), c, k)$ of EDGEMONITORING as follows: G' is a complete graph and for each edge $e \in E'$, we have $c(e) = k - 1$ if $e \in E$ and $c(e) = 0$ otherwise.

We show that (G, k) is a positive instance of INDEPENDENTSET if and only if (G', c, k) is a positive instance of EDGEMONITORING. First of all, notice that there is no monitoring set of size less than k . Indeed, assume, for the sake of contradiction, that there is a monitoring set S of size less than k . Since G is connected, there exists an edge e incident to a vertex in S and such that $c(e) = k - 1$. We have $M(e) \cap S < k - 1$ so there is a contradiction.

Now, let $S \subseteq V$ such that $|S| = k$. Then, we have: S is a monitoring set of (G', c) iff for each $e \in E$, $|S \setminus e| \geq k - 1$ iff for each $e \in E$ in E , $|S \cap e| \leq 1$ iff S is a stable of G .

Now, we show a Turing FPT-reduction from EDGEMONITORING to INDEPENDENTSET. The reduction is presented in Algorithm 1. Notice that this algorithm is recursive.

First, let us prove that (G, c) admits a monitoring set of size at most k if Algorithm 1 returns True. We proceed by induction on k . If $k = 0$, it is clear that Algorithm 1 returns True if and only if $C = 0$. Now, assume that $k > 0$. If Line 6 returns True then (G, c) admits a monitoring set of size at most $k - 1$ by induction hypothesis. Assume now that Line 11 returns True. Then, there

Algorithm 1

Input: $G = (V, E), c, k$

- 1: Let $C = \max\{c(e) : e \in E\}$
 - 2: **if** $C > k$ **then**
 - 3: **return** False
 - 4: **else**
 - 5: **if** Algorithm 1 with parameters $G, c, k - 1$ returns True **then**
 - 6: **return** True
 - 7: **else**
 - 8: Let V^* built from V by removing the extremities of edges e with $c(e) = k$
 - 9: Let $E^* = \{uv \in E : c(uv) = k - 1 \wedge u \in V^* \wedge v \in V^*\}$
 - 10: **if** there exists an independent set of size k in $G^* = (V^*, E^*)$ **then**
 - 11: **return** True
 - 12: **else**
 - 13: **return** False
-

exists an independent set S of size k in G^* . Thus, S is a monitoring set of (G, c) . Indeed (G, c) does not admit an edge e with $c(e) > k$ by Lines 2-3. Edges e with $c(e) = k$ have no extremities in S by construction of G^* . Hence, these edges are monitored by S . Edges e with $c(e) = k - 1$ have at most one extremity in S also by construction of G^* . Thus, these edges are monitored by S . Edges e with $c(e) \leq k - 2$ are necessarily monitored by S since $|S| = k$.

Now, let us prove that Algorithm 1 returns True if (G, c) admits a monitoring set S of size at most k . We proceed by induction on k . If $k = 0$ then necessarily $C = 0$. Thus, Algorithm 1 returns True. Now, assume that $k > 0$. If $|S| \leq k - 1$ then Algorithm 1 returns True in Line 6 by induction hypothesis. Assume now that $|S| = k$ then it is easily seen that S is an independent set of G^* with $|S| = k$. Then Algorithm 1 returns True in Line 11. This completes the proof. \square

Lemma 5. *WEM can be solved in polynomial time on C -bounded weighted complete graphs.*

Proof. Let $(G = (V, E), w, c)$ with G a complete graph. By Lemma 2, $\gamma_m(G, c) \leq C + 2$. Therefore, it suffices to enumerate all sets $S \subseteq V$ that monitor G and such that $|S| \leq C + 2$. There are $O(n^{C+2})$ such sets. Thus, the problem can be computed in polynomial time. \square

Lemma 6. *WEM can be solved in quasi-linear time on uniform complete graphs.*

Proof. Let $(G = (V, E), w, c)$ such that G is a complete graph and c is l -uniform. By Lemma 3, $\gamma_m(G, c) = C + 2$ and by Lemma 2, every set $S \subseteq V$ of size $C + 2$ monitors G . Thus, if we choose S as the set of the $C + 2$ first elements in V sorted by increasing weight, we obtain an optimal solution for $\text{WEM}(G, w, c)$. We only need to sort V which can be done in time $|V| \log |V|$. \square

The following lemma is useful to establish the connection between γ_m of a graph G and γ_m of its 2-connected components.

We denote $\gamma_m(G_1, w, c|u) = \min\{w(S) : S \text{ is a monitoring set of } (G, c) \text{ and } u \in S\}$

Lemma 7. *Let $(G = (V, E), w, c)$ be a weighted graph, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ two graphs and $u \in V$ such that $V = V_1 \cup V_2$, $E = E_1 \cup E_2$ and $V_1 \cap V_2 = \{u\}$. Let $d = \gamma_m(G_1, w, c|u) - \gamma_m(G_1, w, c)$. Let w' obtained from w by replacing the weight of u by d . Then $\gamma_m(G, w, c) = \gamma_m(G_1, w, c) + \gamma_m(G_2, w', c)$.*

Proof. Let S_1, S'_1, S_2 be optimal solutions of $\text{WEM}(G_1, w, c)$, $\text{WEM}(G_1, w, c|u)$, $\text{WEM}(G_2, w', c)$ respectively.

We first prove $\gamma_m(G, w, c) \leq \gamma_m(G_1, w, c) + \gamma_m(G_2, w', c)$: if $u \notin S_2$ then $S_1 \cup S_2$ is a solution of $\text{WEM}(G, w, c)$ having weight $w(S_1) + w'(S_2)$. If $u \in S_2$ then $S'_1 \cup S_2$ is a solution of $\text{WEM}(G, w, c)$ having weight $w(S'_1) + w(S_2) - d = w(S_1) + w'(S_2)$. Thus we have $\gamma_m(G, w, c) \leq \gamma_m(G_1, w, c) + \gamma_m(G_2, w', c)$.

Now we prove $\gamma_m(G, w, c) \geq \gamma_m(G_1, w, c) + \gamma_m(G_2, w', c)$: let S^* be an optimal solution of $\text{WEM}(G, w, c)$. We have $S_1^* = S^* \cap V_1$ and $S_2^* = S^* \cap V_2$ are solutions of $\text{WEM}(G_1, w, c)$ and $\text{WEM}(G_2, w', c)$ respectively. We have to consider two cases:

$u \notin S^*$: We have $w(S_1^*) \geq w(S_1)$ and $w_2'(S_2^*) \geq w_2'(S_2)$ by optimality of S_1 and S_2 . Since $w(S^*) = w(S_1^*) + w(S_2^*)$, $w(S^*) \geq w(S_1) + w(S_2)$.

$u \in S^*$: This implies that $w(S^*) = w(S_1^*) + w'(S_2^*) - d$. Since $w'(S_2^*) \geq w(S_2)$ and $w(S_1^*) \geq w(S'_1)$, then

$$w(S^*) \geq w(S'_1) + w_2'(S_2) - d = w(S_1) + w_2'(S_2)$$

Consequently we have $\gamma_m(G, w, c) \geq \gamma_m(G_1, w, c) + \gamma_m(G_2, w', c)$. This completes the proof of the lemma. \square

Theorem 8. *The two statements hold:*

1. WEM can be solved in polynomial time on C -bounded weighted block graphs.
2. WEM can be solved in quasi-linear time for block graphs $(G = (V, E), w, c)$ where c is uniform.

Proof. Without loss of generality, we can assume that G is connected. We will prove the first statement. The proof of the second statement is similar. Let $(G = (V, E), w, c)$ be a C -bounded weighted block graph. We first compute the block-cut tree T of G . This can be done in linear time [13]. Then, we choose a clique V_1 that corresponds to a leaf of T and u the articulation point that is neighbor of V_1 in T . Let $G_1 = (V_1, E_1) = G[V_1]$ and $G_2 = (V_2, E_2) = G[(V \setminus V_1) \cup \{u\}]$. G_2 is also a block graph. Thus, we can apply Lemma 7. It suffices to compute $\gamma_m(G_1, w, c)$, $\gamma_m(G_1, w, c|u)$ and $\gamma_m(G_2, w', c)$. $\gamma_m(G_1, w, c)$ can be computed in polynomial time by using Lemma 5. Proof of Lemma 5 can be easily modified to compute $\gamma_m(G_1, w, c|u)$. $\gamma_m(G_2, w', c)$ can be computed by induction. \square

4. PTAS for the WEM problem in weighted complete graphs

In this section, we study the approximation complexity of the weighted monitoring set problem in vertex-weighted complete graphs.

Theorem 9. *There exists a PTAS for WEM on complete graphs.*

Proof. Fix $\epsilon > 0$ and $k = \lceil 2/\epsilon \rceil$. Let $G = (V, E), w, c$ such that G is a complete graph and $C = \max\{c(e) : e \in E\}$. Let OPT denote an optimal solution for WEM(G, w, c).

We have to consider three different cases:

Case 1. $C \leq k$:

Using Lemma 2, we have $|\text{OPT}| \leq C + 2 \leq k + 2$. We just need to enumerate all the sets with size at most $k + 2$. We can do it in polynomial time $O(n^{k+2})$.

Case 2. $|V| < C + 2$:

Clearly, there exists no monitoring set for (G, c) since there exists an edge $e = \{u, v\}$ such that $c(e) = C$ and $M(e) < C$.

Case 3. $C \geq k$ and $|V| \geq C + 2$:

Let S_{first} be the set of the first $C + 2$ vertices sorted in ascending order by weight $w(v)$. Let \mathcal{C} be the set of sets $S \subseteq V$ such that $C \leq |S| \leq C + 2$ and $|S \setminus S_{first}| \leq k$. We prove that \mathcal{C} has a polynomial size. Indeed, we have

$$|\mathcal{C}| \leq |\{S \cap S_{first} : S \in \mathcal{C}\}| \times |\{S \setminus S_{first} : S \in \mathcal{C}\}|$$

It holds

$$|\{S \cap S_{first} : S \in \mathcal{C}\}| = \sum_{i=C}^{C+2} \sum_{j=0}^k |\{S \cap S_{first} : S \in V \wedge |S| = i \wedge |S \setminus S_{first}| = j\}| \quad (1)$$

$$= \sum_{i=C}^{C+2} \sum_{j=0}^k \binom{C+2}{i-j} \leq O(C^k) \quad (2)$$

Since $|S \setminus S_{first}| \leq k$ for every $S \in \mathcal{C}$, it holds $|\{S \setminus S_{first} : S \in \mathcal{C}\}| \leq O(n^k)$. Thus $|\mathcal{C}| = O(C^k n^k)$ is polynomial in $|V|$.

The algorithm consists to enumerate all the sets in \mathcal{C} and take a solution of minimum weight. This can be done in polynomial time. We distinguish two subcases as follows:

Case 3.a. $\text{OPT} \in \mathcal{C}$:

Clearly, the algorithm returns an optimal solution.

Case 3.b. $\text{OPT} \notin \mathcal{C}$:

Notice that S_{first} is a (non necessary optimal) solution by Lemma 2 and the algorithm returns a solution S such that $w(S) \leq w(S_{first})$. We will prove that $w(S_{first}) \leq (1 + \epsilon)w(\text{OPT})$. Let a_1, \dots, a_l denote the vertices in $\text{OPT} \cap S_{first}$. Let b_1, \dots, b_m denote the vertices in $\text{OPT} \setminus S_{first}$. Let c_1, \dots, c_n denote

the vertices in $S_{first} \setminus \text{OPT}$ sorted in ascending order by weight $w(v)$. Since $|\text{OPT} \setminus S_{first}| \geq k$, we have $m \geq k$. In the following, we will bound the approximation ratio of the solution:

$$\frac{w(S_{first})}{w(\text{OPT})} = \frac{w(a_1) + \dots + w(a_l) + w(c_1) + \dots + w(c_n)}{w(a_1) + \dots + w(a_l) + w(b_1) + \dots + w(b_m)} \quad (3)$$

$$\leq \frac{w(c_1) + \dots + w(c_n)}{w(b_1) + \dots + w(b_m)} \quad (4)$$

$$\leq \frac{n \cdot w(c_n)}{m \cdot w(c_n)} = \frac{n}{m} \quad (5)$$

$$\leq \frac{m+2}{m} \quad (6)$$

$$\leq \frac{k+2}{k} = 1 + \frac{2}{k} \quad (7)$$

$$\leq 1 + \frac{2}{\lceil \frac{2}{\epsilon} \rceil} \leq 1 + \epsilon \quad (8)$$

In (3), we use the fact that if $a, b, c > 0$ and $b \geq c$ then $\frac{a+b}{a+c} \leq \frac{b}{c}$. Since $w(\text{OPT}) \leq w(S_{first})$, we obtain $w(c_1) + \dots + w(c_n) \geq w(b_1) + \dots + w(b_m)$. Thus, we get (4). We obtain (5) since $w(c_i) \leq w(c_n)$ for any $i \in [1, n]$ and $w(b_i) \geq w(c_n)$ for any $i \in [1, m]$. To get (6), we use the property that $|\text{OPT}| \geq C$ and $|S_{first}| = C + 2$. Since $m \geq k$ and $k = \lceil 2/\epsilon \rceil$, the rest follows. \square

5. Interval graphs

In this section, we give a polynomial algorithm for computing WEM on weighted interval graphs. This algorithm uses dynamic programming. First, we introduce some definitions.

A graph $G = (V, E)$ is an interval graph if there exists $|V|$ intervals $(I_i)_{i \in V} = ([a_i, b_i])_{i \in V}$ of the real line such that $\{i, j\} \in E$ if and only if $I_i \cap I_j \neq \emptyset$ for every distinct vertices $i, j \in V$. We say that $(I_i)_{i \in V}$ is a realization of G . Without loss of generality, we can assume that there are no intervals I_i and I_j that have a common extremity.

Given an interval graph $G = (V, E)$ and a realization $(I_i)_{i \in V}$, we define a total order $<_L$ (resp. $<_R$) over V such that $i <_L j$ (resp. $i <_R j$) if $a_i < a_j$ (resp. $b_i < b_j$).

The following definition is a refinement of the nice tree decomposition introduced by Kloks [15]

Definition 1. [10] Let $G = (V, E)$ be an interval graph and $(I_i)_{i \in V}$ be a realization of G . A nice path decomposition of G is a sequence of sets of vertices B_0, \dots, B_l such that

- all sets B_i are cliques of G ;
- every edge $e \in E$ appears in a set B_i ,
- for every vertex $v \in E$, the set of indices i such that $v \in B_i$ is a segment of $[0, l]$.
- $B_0 = \emptyset$ and $B_l = \emptyset$;
- For every $i \in [1, l]$,
 - $B_i = B_{i-1} \cup \{v\}$ (i introduces the vertex v)
 - or $B_i = B_{i-1} \setminus \{v\}$ (i forgets the vertex v).
- the order in which vertices are introduced corresponds to $<_L$
- the order in which vertices are forgotten corresponds to $<_R$

Lemma 10. [10] Let $G = (V, E)$ be an interval graph and $(I_i)_{i \in V}$ be a realization of G . Then G has a nice path-decomposition that can be computed in linear time.

For the next lemmas, we consider an interval graph $G = (V, E)$ and a nice path-decomposition B_0, \dots, B_l of G . Moreover, we introduce the following notations. For $i \in [0, l]$, F_i is the set of vertices appearing in some set B_j , $j < i$, but not in B_i . $V_i = F_i \cup B_i$ and $G_i = G[V_i]$.

A set $S \subseteq V_i$ is an i -partial solution if every edge e in G_i that has an extremity in F_i is $c(e)$ -monitored by S . The i -representant W of $S \subseteq V_i$, denoted by $\text{repr}_i(S)$, contains exactly the $C + 2$ greatest vertices in $S \cap N[B_i]$ w.r.t. $<_R$ or is $S \cap N[B_i]$ if $|S \cap N[B_i]| < C + 2$. We say that S extends W if W is the i -representant of S .

We denote by \mathcal{F}_i^* the set of i -representants of i -partial solutions. $w_i^* : \mathcal{F}_i^* \rightarrow \mathbb{Q}^+$ is a function such that $w_i^*(W) = \min\{w(S) : S \text{ is an } i\text{-partial solution that extends } W\}$.

Before presenting the algorithm, we introduce two lemmas. The second is the key of the algorithm.

Lemma 11. Let $u \in B_i$, $v_1, v_2 \in V_i$ such that $v_1 <_R v_2$ and $v_1 \in N[u]$. Then $v_2 \in N[u]$.

Proof. Let $[a_u, b_u]$, $[a_{v_1}, b_{v_1}]$ and $[a_{v_2}, b_{v_2}]$ the intervals that represent u , v_1 and v_2 respectively in the realization of G . Since $u \in B_i$ and $v_2 \in V_i$, $b_u > a_{v_1}$ and $b_u > a_{v_2}$. Since $v_1 \in N[u]$, we have $a_u < b_{v_1}$ and since $v_1 <_R v_2$, we have $a_u < b_{v_2}$. Thus $[a_u, b_u] \cap [a_{v_2}, b_{v_2}] \neq \emptyset$. Consequently, $v_2 \in N[u]$. \square

Lemma 12. Let $S \subseteq V_i$, $W = \text{repr}_i(S)$, $v_1, v_2 \in B_i$ and $\alpha \in [0, C]$. If $\{v_1, v_2\}$ is α -monitored by S then $\{v_1, v_2\}$ is α -monitored by W .

Proof. First, notice that $M(\{v_1, v_2\}) \subseteq N[B_i]$. If $|S \cap N[B_i]| \leq C + 2$, then $W = S \cap N[B_i]$ and the lemma is trivially verified. Now, assume that $|S \cap N[B_i]| > C + 2$ and let $u \in (S \setminus W) \cap M(\{v_1, v_2\})$. By Lemma 11, every vertex $u' \in W$ belongs to $N[v_1]$ and $N[v_2]$. So all elements in W except at most two (v_1 and v_2) belong to $M(\{v_1, v_2\})$. Thus $|M(\{v_1, v_2\}) \cap W| \geq C$ and $\{v_1, v_2\}$ is α -monitored by W . \square

To solve WEM on interval graphs, a naive algorithm consists to iterate over the sets B_i and to compute for each i the set of i -partial solutions. Unfortunately, the algorithm is non polynomial since the set of i -partial solutions can be exponential. The key of the algorithm is as follows: instead of considering all the i -partial solutions, we consider the representants of the i -partial solutions. Since the number of representants is polynomially bounded by $|V|$, the algorithm will run in polynomial time. Lemma 12 guarantees that we don't miss solutions. Indeed, let S be an i -partial solution. If $i + 1$ introduces the node v , then S and $S \cup \{v\}$ are $(i + 1)$ -partial solutions. If $i + 1$ forgets the node v then S is an $(i + 1)$ -partial solution if and only if every forgotten edge e (i.e. an edge having v as extremity and the other extremity in B_{i+1}) is $c(e)$ -monitored by S . But thanks to Lemma 12, it suffices to check that these edges are $c(e)$ -monitored by $\text{repr}_i(S)$.

We present now Algorithm 2.

The next lemma shows that the sets \mathcal{F}_i and functions w_i computed by Algorithm 2 correspond to the sets \mathcal{F}_i^* and functions w_i^* defined previously.

Lemma 13. *For every $i \in [0, l]$, after the run of Algorithm 2, we have $\mathcal{F}_i = \mathcal{F}_i^*$ and $w_i(S) = w_i^*(S)$ for every $S \in \mathcal{F}_i$.*

Proof. We prove by induction on i . The property is clearly verified for $i = 0$. Now, suppose that the property holds for i and prove it for $i + 1$.

$\mathcal{F}_{i+1} \subseteq \mathcal{F}_{i+1}^*$ and for each $W \in \mathcal{F}_{i+1}$, $w_{i+1}^*(W) \leq w_{i+1}(W)$: let $W' \in \mathcal{F}_{i+1}$. We consider two cases.

$i + 1$ forgets the vertex v : then, W' comes from some $W \in \mathcal{F}_i$ such that $W' = \text{repr}(W)$, $w_{i+1}(W') = w_i(W)$ and W' is added to \mathcal{F}_{i+1} by Lines 9-11. Using the induction hypothesis, $W \in \mathcal{F}_i^*$ and $w_i(W) = w_i^*(W)$. Let S be a i -partial solution of weight $w(S) = w_i^*(W)$ that extends W . By Line 8 of the algorithm, every edge $e = \{u, v\}$ where $u \in B_i$ is $c(e)$ -monitored by W and thus by S . Consequently, S is an $(i + 1)$ -partial solution with $\text{repr}_{i+1}(S) = \text{repr}_{i+1}(W) = W'$. Thus, $W' \in \mathcal{F}_{i+1}^*$ and $w_{i+1}^*(W') \leq w(S) = w_i^*(W) = w_i(W) = w_{i+1}(W')$.

$i + 1$ introduces the vertex v : There are two possibilities.

$v \notin W'$: then W' comes from some $W \in \mathcal{F}_i$ such that $W' = \text{repr}(W)$, $w_{i+1}(W') = w_i(W)$ and W' is added to \mathcal{F}_{i+1} by Lines 14-16. By induction hypothesis, $W \in \mathcal{F}_i^*$ and $w_i(W) = w_i^*(W)$. Let S be a i -partial solution of weight $w(S) = w_i^*(W)$ that extends W . S is an $(i + 1)$ -partial solution with $\text{repr}_{i+1}(S) = \text{repr}_{i+1}(W) = W'$. Thus $W' \in \mathcal{F}_{i+1}^*$ and $w_{i+1}^*(W') \leq w(S) = w_i^*(W) = w_i(W) = w_{i+1}(W')$.

$v \in W'$: W' comes from some $W \in \mathcal{F}_i$ such that $W' = \text{repr}(W + \{v\})$, $w_{i+1}(W') = w_i(W) + w(v)$ and W' is added to \mathcal{F}_{i+1} by Lines 17-19. Let S be a

Algorithm 2 Algorithm for WEM on interval graphs

Input: a weighted interval graph (G, c, w) and a nice path decomposition

B_0, \dots, B_l

- 1: $\mathcal{F}_0 \leftarrow \{\emptyset\}$
- 2: $w_0(\emptyset) = 0$
- 3: **for** i from 1 to l **do**
- 4: $\mathcal{F}_i \leftarrow \emptyset$
- 5: $w_i(S) = +\infty$ for any S
- 6: **if** i forgets the node v **then**
- 7: **for** $W \in \mathcal{F}_{i-1}$ **do**
- 8: **if** every edge $e = \{u, v\}$ with $u \in B_i$ is $c(e)$ -monitored by W **then**
- 9: $W' \leftarrow \text{repr}_i(W)$
- 10: $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{W'\}$
- 11: $w_i(W') \leftarrow \min\{w_i(W'), w_{i-1}(W)\}$
- 12: **else if** i introduces the node v **then**
- 13: **for** $W \in \mathcal{F}_{i-1}$ **do**
- 14: $W' \leftarrow \text{repr}_i(W)$
- 15: $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{W'\}$
- 16: $w_i(W') \leftarrow \min\{w_i(W'), w_{i-1}(W)\}$
- 17: $W' \leftarrow \text{repr}_i(W \cup \{v\})$
- 18: $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \{W'\}$
- 19: $w_i(W') \leftarrow \min\{w_i(W'), w_{i-1}(W) + w(v)\}$
- 20: **if** $\mathcal{F}_l = \emptyset$ **then**
- 21: **return** $+\infty$
- 22: **else**
- 23: **return** $\min\{w_l(W) : W \in \mathcal{F}_l\}$

i -partial solution of weight $w(S) = w_i^*(W)$ that extends W . $S' = S \cup \{v\}$ is an $(i+1)$ -partial solution with $\text{repr}_{i+1}(S \cup \{v\}) = \text{repr}_{i+1}(W \cup \{v\}) = W'$. Thus $W' \in \mathcal{F}_{i+1}^*$ and $w_{i+1}^*(W') \leq w(S + \{v\}) = w_i^*(W) + w(v) = w_i(W) + w(v) = w_{i+1}(W')$.

$\mathcal{F}_{i+1}^* \subseteq \mathcal{F}_{i+1}$ and for each $W \in \mathcal{F}_{i+1}^*$, $w_{i+1}(W') \leq w_{i+1}^*(W')$: let $W' \in \mathcal{F}_{i+1}^*$ and S' be an $(i+1)$ -partial solution that extends W and such that $w(S') = w_{i+1}^*(W')$. We also consider two cases:

$i+1$ forgets the vertex v : then S' is an i -partial solution. Let $W = \text{repr}_i(S')$. Then $W' = \text{repr}_{i+1}(W)$. Using the induction hypothesis, $W \in \mathcal{F}_i$ and $w_i(W) = w_i^*(W)$. By definition of a $(i+1)$ -partial solution, every edge $e = \{u, v\}$ with $u \in B_i$ is $c(e)$ -monitored by S' . By applying Lemma 12, these edges are also $c(e)$ -monitored by W . Thus, Line 8 of the algorithm succeeds and $W' = \text{repr}_{i+1}(W)$ is added to \mathcal{F}_{i+1} and by Line 10 $w_{i+1}(W') \leq w_i(W) = w_i^*(W) = w(S') = w_{i+1}^*(W')$.

$i+1$ introduces the vertex v . There are two possibilities.

$v \notin S'$: then S' is an i -partial solution. Let $W = \text{repr}_i(S')$. Using the induction hypothesis, $W \in \mathcal{F}_i$ and $w_i(W) = w_i^*(W)$. Thus, $W' = \text{repr}_{i+1}(W)$ is added to \mathcal{F}_{i+1} by Line 14 and by Line 15 $w_{i+1}(W') \leq w_i(W) = w_i^*(W) = w(S') = w_{i+1}^*(W')$.

$v \in S'$: let $S = S' - v$. Then S' is an i -partial solution. Let $W = \text{repr}_i(S)$. Using the induction hypothesis, $W \in \mathcal{F}_i$ and $w_i(W) = w_i^*(W)$. Thus, $W' = \text{repr}_{i+1}(S \cup \{v\}) = \text{repr}_{i+1}(W \cup \{v\})$ is added to \mathcal{F}_{i+1} by Line 18-19 and $w_{i+1}(W') \leq w_i(W) + w(v) = w_i^*(W) + w(v) = w(S) + w(v) = w(S') = w_{i+1}^*(W')$. \square

Theorem 14. WEM on C -bounded weighted interval graphs is in P. More precisely, it can be solved in time $O(|V|^{C+4})$.

Proof. Thanks to Lemma 13, it is clear that Algorithm 2 is exact. Let prove that it runs in the expected time. The algorithm consists of a main loop that does $|V| + 1$ iterations. Within this loop, we have two possibilities: forgetting or introducing a vertex. In the two cases, we loop over the elements of \mathcal{F}_{i-1} . Each step of the loop can be done in time $O(N(B_i))$ (since C is bounded) in both cases. The size of \mathcal{F}_{i-1} is bounded by $(N[B_{i-1}] \cap V_{i-1})^{C+2}$. Therefore the time spent within a step of the main loop is $O((N[B_{i-1}] \cap V_{i-1})^{C+3})$. Since $N[B_{i-1}] \cap V_{i-1}$ is bounded by $|V|$, Algorithm 5 runs in time $O(|V|^{C+4})$. \square

The complexity of the algorithm can be refined in the case of unit interval graphs.

Lemma 15. Let C be a clique of an unit interval graph $G = (V, E)$. Then $N[C] \leq 3\omega(G)$.

Proof. Let $(I_i)_{i \in E}$ be a realization of G . Since G is an unit interval graph, we have $u \leq_L v \Leftrightarrow u \leq_R v$ for every $x, y \in V$. For every vertex $v \in V$, we denote by $N_{\leq}[v]$ (resp. $N_{\geq}[v]$) the set $\{u : u \in N[v] \wedge u \leq_L v\}$ (resp. $\{u : u \in N[v] \wedge u \geq_L v\}$). Let v_{min} (resp. v_{max}) be the minimal (resp. maximal) vertex of C w.r.t

\leq_L . It is easily seen that $N[C] = N_{\leq}[v_{min}] \cup (N_{\geq}[v_{min}] \cap N_{\leq}[v_{max}]) \cup N_{\geq}[v_{max}]$ and that $N_{\leq}[v_{min}]$, $N_{\geq}[v_{min}] \cap N_{\leq}[v_{max}]$ and $N_{\geq}[v_{max}]$ are clique of G . Thus $N[C] \leq 3\omega(G)$.⁹ □

Theorem 16. WEM can be solved in time $O(\omega(G)^{C+3}|V|)$ on C -bounded weighted unit interval graphs.

Proof. We refine the running time of Theorem 14. Thanks to Lemma 15, we can bound $N[B_{i-1}] \cap V_{i-1}$ by $3\omega(G)$. Thus, we deduce that the overall running time is $O(\omega(G)^{C+3}|V|)$ in weighted unit interval graphs. □

6. Cographs

Let $G_1 = (V_1, E_2)$ and $G_2 = (V_2, E_2)$ such that $V_1 \cap V_2 = \emptyset$. The join of G_1 and G_2 is the graph $G = (V_1 \cup V_2, E_1 \cup E_2 \cup \{\{u, v\} : u \in V_1 \wedge v \in V_2\})$. The class of cographs is defined by induction.

- The graph which contains one vertex is a cograph;
- The (disjoint) union and the join of two cographs are cographs.

Lemma 17. Let $G = (V, E)$ be the join of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Let S be a total dominating set of G_1 . Then, S monitors all edges between V_1 and V_2 .

Proof. Let $\{u, v\}$ be an edge between G_1 and G_2 such that $u \in V_1$. Then there exists a vertex $u_1 \in S$ adjacent to u . Thus, $\{u, v\}$ is monitored by S since $\{u, v, u_1\}$ is a triangle of G . □

Lemma 18. Let $G = (V, E)$ be the join of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Let S be a monitoring set of G . Then $S \cap V_1$ is a total dominating set of G_1 or $S \cap V_2$ is a total dominating set of G_2 .

Proof. Assume for the sake of contradiction that S_1 is not a total dominating set of G_1 and S_2 is not a total dominating set of G_2 . Then there exists an edge $\{u, v\} \in E$ such that u has no neighbor in S_1 and v has no neighbor in S_2 . Thus, $\{u, v\}$ is not monitored by S . □

Lemma 19. Let G be the join of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Let S be a minimal monitoring set of G . Then $|S \cap V_1| \leq 1$ or $|S \cap V_2| \leq 1$.

Proof. Let S be a minimal monitoring set of G , $S_1 = S \cap V_1$ and $S_2 = S \cap V_2$. Assume, for the sake of contradiction, that $|S_1| \geq 2$ and $|S_2| \geq 2$. By Lemma 18, S_1 is a total dominating set of G_1 or S_2 is a total dominating set of G_2 . By symmetry, suppose that S_1 is a total dominating set of G_1 . Then S_1 monitors all edges between V_1 and V_2 by Lemma 17 and all edges in V_2 . Consequently, for every vertex $u \in V_2$, $S_1 \cup \{u\}$ is a monitoring set of G since u monitors all edges in V_1 . Thus, S is not minimal. □

Lemma 20. *Let $G = (V, E)$ be a graph with no isolated vertices and S a monitoring set of G . Then, S is a total dominating set of G .*

Proof. Let v be a vertex in V . Since G has no isolated vertices, there is a vertex $e = (v, v_1)$ incident to v . Since S is a monitoring set of G , there is a vertex $v_2 \in S$ such that $\{v, v_1, v_2\}$ is a triangle in G . Thus, v is adjacent to a vertex in S . \square

Combining Lemmas 17, 18 and 20, we obtain the following lemma.

Lemma 21. *Let $G = (V, E)$ be the join of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Let $S_1 = S \cap V_1$ and $S_2 = S \cap V_2$. The two statements hold.*

- *If $S_1 \neq \emptyset$ and $S_2 \neq \emptyset$, then S is a monitoring set of G if and only if S_1 is a total dominating set of G_1 or S_2 is a total dominating set of G_2 .*
- *If $S_2 = \emptyset$ (resp. $S_1 = \emptyset$), then S is a monitoring set of G if and only if G_1 (resp. G_2) has no isolated vertices and S_1 (resp. S_2) is a monitoring set of G_1 (resp. G_2).*

The following lemma is a direct consequence of Lemma 19 and Lemma 21.

Lemma 22. *Let $G = (V, E)$ be a graph. If G is the (disjoint) union of two graphs G_1 and G_2 . Then,*

$$\gamma_m(G, w) = \gamma_m(G_1, w) + \gamma_m(G_2, w)$$

If G is the join of two graphs G_1 and G_2 .

$$\gamma_m(G, w) = \min \begin{cases} \gamma_t(G_1, w) + \min\{w(v) : v \in V_2\} \\ \min\{w(v) : v \in V_1\} + \gamma_t(G_2, w) \\ \gamma_m(G_1, w) \text{ if } G_1 \text{ has no isolated vertices} \\ \gamma_m(G_2, w) \text{ if } G_2 \text{ has no isolated vertices} \end{cases}$$

Lemma 22 combined with the fact that a cotree is computable in linear time [12] give us a linear time algorithm to compute 1-uniform WEM on cographs.

Theorem 23. *1-uniform WEM can be solved in linear time on cographs.*

7. Split graphs and comparability graphs

A graph $G = (V, E)$ is a split graph if V can be partitioned into C and I where C is a clique of G and I is an independent set of G .

Lemma 24. *Let $G = (V = C \cup I, E)$ be a split graph with minimum degree $\delta(G) \geq 2$ and such that $|C| \geq 3$ and Then, there exists a minimum 2-tuple dominating set (resp. monitoring set) $S \subseteq C$.*

Proof. Let S be a set that minimizes $|S \cap I|$ among all minimum 2-tuple dominating sets of G . For the sake of contradiction, suppose $S \cap I$ non empty and let v be a vertex in $S \cap I$. If $N(v) \subseteq S$, then $S - v$ is also a 2-tuple dominating set of G . Thus, G is not minimum. Otherwise, let $u \in S \setminus N(v)$. Then $S' = S \cup \{u\} - v$ is a minimum 2-tuple dominating set of G with $|S' \cap I| < |S \cap I|$. Thus S does not minimize $|S \cap I|$.

The proof for monitoring sets is quite similar to the proof for 2-tuple dominating sets. Let S be a set that minimizes $|S \cap I|$ among all minimum monitoring sets of G . For the sake of contradiction, suppose $S \cap I$ non empty and let v be a vertex in $S \cap I$. S contains at least 3 vertices and $|S \cap C| \geq 2$. Otherwise, S does not monitor all vertices between C and I . If $N(v) \subseteq S$ and $|S \cap C| \geq 3$, then $S - v$ is also a monitoring set of G . Thus S is not minimum. If $N(v) \subseteq S$ and $|S \cap C| = 2$ then choose a vertex $u \in C \setminus S$. Thus, $S' = S \cup \{u\} - v$ is a minimum monitoring set with $|S' \cap I| < |S \cap I|$. Now, suppose that $N(v) \not\subseteq S$ and let $u \in N(v) \setminus S$. Then, $S' = S \cup \{u\} - v$ is a minimum monitoring set with $|S' \cap I| < |S \cap I|$. That contradicts our assumption. \square

Lemma 25. *Let $G = (V = C \cup I, E)$ be a split graph with minimum degree $\delta(G) \geq 2$ and such that $|C| \geq 3$ and $\gamma_{\times 2}(G) \geq 3$. Then, $\gamma_m(G) = \gamma_{\times 2}(G)$.*

Proof. To see that $\gamma_{\times 2}(G) \leq \gamma_m(G)$, consider a monitoring set S and a vertex v . Since $\delta(G) \geq 2$, v admits a neighbor v_1 . Since $\{v, v_1\}$ is monitored by S , v admits a neighbor $v_2 \in S$ and, since $\{v, v_2\}$ is monitored by S , v admits another neighbor $v_3 \in S$. Thus S is a double dominating set of G .

We will prove that $\gamma_{\times 2}(G) \geq \gamma_m(G)$. Let S be a minimum 2-tuple dominating set of G . Thanks to Lemma 24, we can assume without loss of generality that $S \subseteq C$. Since $|S| \geq 3$, S monitors all edges in $G[C]$. Let $\{u, v\}$ be an edge in G such that $u \in C$ and $v \in I$. Since S dominates twice the vertex v , there is a node $u' \in S \cap N(v)$ distinct to u . Thus $\{u, v\}$ is monitored by u' . Consequently, S is a monitoring set of G . \square

Since 2-tuple domination is NP-complete on split graphs even with these restrictions [16], we obtain the following result.

Theorem 26. *1-uniform EDGEMONITORING is NP-complete on split graphs.*

A graph $G = (V, E)$ is a comparability graph if there exists a poset \leq over V such that $\{x, y\} \in E$ if and only if $x \leq y$ or $y \leq x$ for every $x, y \in E$.

Theorem 27. *1-uniform EDGEMONITORING is NP-complete on comparability graphs.*

Proof. We do a reduction from TOTALDOMINATINGSET on bipartite graphs which has been proved NP-complete [18]. Let $G = (V, E)$ be a bipartite graph. Without loss of generality, assume that G has no isolated vertices. Let G' be the graph obtained from G by adding an universal vertex u . It is clear that G' is a comparability graph. We will prove that $\gamma_m(G') = \gamma_t(G) + 1$. Let S be a

total dominating set of G . Then, $S \cup \{u\}$ is a monitoring set of G . Indeed, every edge in E is covered by u and for every edge $\{u, v\}$ with $v \in V$, there is a vertex $v' \in N(v) \cap S$. Thus, $\{u, v\}$ is monitored by v' . Now, let S be a monitoring set of G' . Then, $u \in S$ because u is the only vertex that monitors edges in E . $S - u$ is a total dominating set of G . Indeed, let v be a vertex in V . $\{u, v\}$ is an edge of G' monitored by a vertex $v' \in S - u$ distinct from v . Thus, v is dominated by v' . \square

8. Planar graphs and unit disk graphs

8.1. Negative results

A graph $G = (V, E)$ is an unit disk graph if there exists a map $f : V \rightarrow \mathbb{R}^2$ satisfying

$$\{u, v\} \in E \Leftrightarrow \|f(u) - f(v)\| \leq 2$$

f is called a geometric representation of G .

Recognizing whether a graph G is an unit disk graph is NP-hard [4]. Thus, computing a geometric representation of an unit disk graph is also NP-hard. Consequently, we suppose that an unit disk graph G is given with a geometric representation f .

Dong et al [7] prove that k -uniform EDGE-MONITORING is NP-complete on unit disk graphs for every $k \geq 2$. We prove a stronger result for 1-uniform EDGE-MONITORING.

Theorem 28. *1-uniform EDGE-MONITORING is NP-complete on planar unit disk graphs given with a geometric representation.*

The proof is inspired by Theorem 4.1 in [6]. As in [6] we use the following lemma:

Lemma 29. [19] *A planar graph G with maximum degree 4 can be embedded in the plane using $O(|V|)$ area in such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of horizontal or vertical segments.*

Proof. (of Theorem 28) We show a reduction from PLANAR-VERTEX-COVER with maximum degree 3 which is NP-complete [11]. Let $G = (V, E)$ be a planar graph with maximum degree 3. Let $\{e_1, \dots, e_{|E|}\}$ be the edges in G . Let $N > 0$ be a sufficient large integer. We draw G in the plane using Lemma 29 (see Figure 2) and we multiply each coordinate by N i.e. each vertex is at coordinate (iN, jN) for some integers i and j . We build $G' = (V', E')$ from G by replacing each edge $e_i = \{u, v\}$ with a subgraph G_{e_i} of vertices $\{a_{i,0} = u, b_{i,0}, b'_{i,0}, a_{i,1}, b_{i,1}, b'_{i,1}, \dots, a_{i,2n_i}, b_{i,2n_i}, b'_{i,2n_i}, a_{i,2n_i+1} = v\}$ where each n_i is an integer that depends on the length of the embedding of e_i . For each $i \in [0, 2n_i]$, we connect b_i and b'_i to a_i and a_{i+1} and we connect b_i to b'_i (see Figure 3).

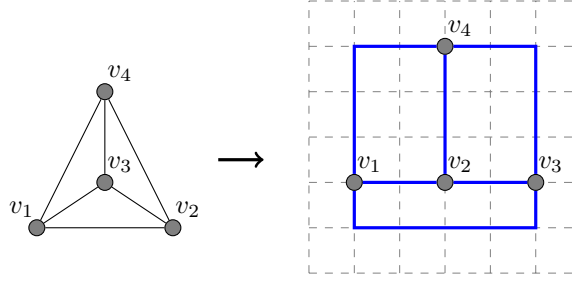


Figure 2: A representation of K_4 in the grid

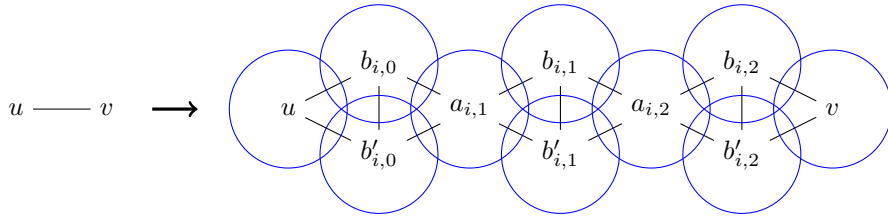


Figure 3: An edge $e_i = \{u, v\}$ and its associate graph G_{e_i} for $n_i = 1$

It is easily seen that the obtained graph G' is planar and that there exists a unit disk representation of G' for N sufficient large. Now, we prove that G admits a vertex-cover S such that $|S| \leq k$ if and only if G' has a monitoring set S' such that $|S'| \leq k' = k + \sum_{i \in [1, |E|]} (5n_i + 2)$. Let A be the set of vertices $a_{i,j}$ for $i \in [1, |E|]$ and $j \in [1, 2n_i]$. Let B be the set of vertices $b_{i,j}$ and $b'_{i,j}$ for $i \in [1, |E|]$ and $j \in [0, 2n_i]$. Clearly, V' is the disjoint union of V , A and B . Moreover, $|B| = \sum_{e_i \in E} (4n_i + 2)$. The proof is an immediate consequence of these three facts.

(1) If a set $S \subseteq V'$ monitors G' then $B \subseteq S$: otherwise, there exists a vertex $b_{i,j}$ or $b'_{i,j}$ that is not in S . Then $\{a_{i,j}, b_{i,j}\}$ or $\{a_{i,j}, b'_{i,j}\}$ is not monitored by S .

(2) Let S be a vertex-cover of G . Then there is a set $A' \subseteq A$ such that $|V(G_{e_i}) \cap A'| = n_i$ for every $i \in [1, |E|]$ and such that $S \cup A' \cup B$ is a monitoring set of G' : let $e_i = \{u, v\}$ be an edge in G . If $u \in S$, then we choose a_{2i} for $i \in [1, n_i]$ as elements of A' . Otherwise ($v \in S$), we choose a_{2i+1} for $i \in [0, n_i - 1]$. It is easily seen that $S \cup A' \cup B$ is a monitoring set of G' .

(3) There exists a minimum monitoring set S of G' such that $V \cap S$ is a vertex-cover of G and $|V(G_{e_i}) \cap A \cap S| = n_i$ for every $i \in [1, |E|]$: assume that $V \cap S$ is not a vertex cover of G . Let $e_i = \{u, v\}$ be an edge in G not covered by $V \cap S$. Then, it is easily seen that $|V(G_{e_i}) \cap A \cap S| > n_i$. Otherwise, an edge $\{b_{i,j}, b'_{i,j}\}$ for some j is not covered by S . Thus, we can replace these vertices by u and n_i vertices in $V(G_{e_i} \cap A$ which monitors every edge $\{b_{i,j}, b'_{i,j}\}$. By iterating this process on every edge in G , we obtain a set S' with the desired properties. Now, assume that $V \cap S$ is a vertex cover of G but there is some i such that

$|V(G_{e_i}) \cap A \cap S'| \neq n_i$. It is easily seen that $|V(G_{e_i}) \cap A \cap S'| < n_i$ implies that an edge $\{b_{i,j}, b'_{i,j}\}$ for some j is not covered by S' and $|V(G_{e_i}) \cap A \cap S'| > n_i$ implies that S' is not minimum. \square

8.2. A PTAS for planar graphs

Now, we introduce a PTAS for planar graphs and fore more general graph classes: apex-minor-free families of graphs.

An apex graph is a graph G such that for some vertex v , $G - v$ is planar. A minor of a graph G is graph that can be obtained from G by a serie of vertex deletions, edge deletions and edge contractions. Given a graph H , a family of graphs \mathcal{F} is H -minor-free if H is not a minor of any graph $G \in \mathcal{F}$. A family of graphs \mathcal{F} is apex-minor-free if it is H -minor-free for some apex graph H . A minor-closed family \mathcal{F} of graphs has bounded local treewidth if there is some function f such that every graph in \mathcal{F} with diameter d has treewidth at most $f(d)$.

In our proof, we use this fundamental property.

Theorem 30. [8] *Let \mathcal{F} be a minor-closed family of graphs. Then \mathcal{F} has bounded local treewidth iff \mathcal{F} is apex-minor-free.*

We also need, the following result of Baste and al.

Theorem 31. [3] *WEM is solvable in time in time $2^{O(tw^2 \log C)}|V|$ where tw is the treewidth of G .*

Notice that the proof of this theorem in [3] does not consider weights on vertices but we can easily generalize it.

Now, We can prove our theorem.

Theorem 32. *There exists a PTAS for WEM on any weighted apex-minor-free families of graphs.*

Proof. Let H be an apex graph. Without loss of generality, we consider WEM on the maximal minor-closed family of graphes \mathcal{F} that excludes the graph H . Thus, \mathcal{F} has bounded local treewidth. We use the classical Baker's technique [2] on planar graphes generalized by Eppstein [8] on bounded local treewidth families of graphs. Let (G, w, c) be a weighted graph with $G \in \mathcal{F}$. Without loss of generality, we assume that G is connected. We choose an arbitrary vertex $v \in V$ and we define L_i as set of vertices at distance i from v . L_i is called the layer of level i . Let l be the maximal distance between v and a vertex of G . These layers can be obtained in linear time by breadth first search. The key idea is that, since \mathcal{F} has bounded local treewidth, the graph induced by k consecutive layers L_i, \dots, L_{i+k-1} has a treewidth bounded by $f(k+1)$ ¹.

¹the supergraph obtained from G by removing all layers L_j with $j \geq i+k$ and by contracting all layers L_j with $j < i$ into one vertex belongs to \mathcal{F} and has a diameter at most $k+1$. Thus its treewidth is at most $f(k+1)$

Another important point is that every edge of G has extremities in the same layer or in two consecutive layers.

Fix $\epsilon > 0$ and k such that $\frac{k+2}{k} \leq 1 + \epsilon$. We will give a $\frac{k+2}{k}$ -approximation algorithm that is polynomial for a fixed k . We define B_i as the union of k consecutive layers $L_i \cup \dots \cup L_{i+k-1}$ ² and R_i as the union of $k+2$ consecutive layers $L_{i-1} \cup \dots \cup L_{i+k}$. Let P_i be the subproblem whose output is a set S of minimum weight in R_i that monitors all edges having at least one extremity in B_i . Since $G[R_i]$ has treewidth at most $f(k+1)$, we can solve this problem in polynomial time using Theorem 31 by replacing the weight $c(e)$ of edges e having both extremities outside B_i with 0.

Now, we present Algorithm 3 that is a PTAS for WEM on apex-minor-free families of graphs.

Algorithm 3 PTAS for WEM on an apex-minor-free family of graphs

Input: (G, c, w) , $\epsilon > 0$

- 1: let k such that $\frac{k+2}{k} \leq 1 + \epsilon$
- 2: **if** there exists an edge $e \in E$ such that $c(e) > M(e)$ **then**
- 3: **return** False
- 4: **for** i from 0 to $k-1$ **do**
- 5: **for** j from -1 to $\lceil \frac{i}{k} \rceil$ **do**
- 6: let $S_{i,j}$ be an optimal solution of P_{i+kj}
- 7: let $S_i = S_{i,-1} \cup \dots \cup S_{i, \lceil \frac{i}{k} \rceil}$
- 8: let S be a set S_i such that $w(S_i)$ is minimal
- 9: **return** S

It is clear that Algorithm 3 runs in polynomial time when ϵ is fixed. Let us prove that Algorithm 3 is correct. First, notice that there exists a monitoring set of (G, c) if and only if Line 2 of Algorithm 3 fails. Now, assume that (G, c) admits a monitoring set and let OPT be an optimal solution for WEM(G, c, w). Notice that, for any i , S_i is a (not necessarily optimal) monitoring set of (G, c) . Thus, S is also a monitoring set of (G, c) .

Besides, $\text{OPT} \cap R_i$ is a (not necessarily optimal) solution of P_i . Indeed, an edge that have an extremity in B_i can only be monitored by vertices in R_i . Consequently, for any i and j , it holds that $w(S_{i,j}) \leq w(\text{OPT} \cap R_{i+kj})$. Therefore, for any i , we have

$$w(S_i) \leq \sum_{j=-1}^{\lceil \frac{i}{k} \rceil} w(S_{i,j}) \leq \sum_{j=-1}^{\lceil \frac{i}{k} \rceil} w(\text{OPT} \cap R_{i+kj})$$

There exists an integer $i \in [0, k-1]$ such that $w(\text{OPT} \cap (C_i \cup C_{i+1})) \leq \frac{2}{k} w(\text{OPT})$ where C_i is the union of layers $L_{i'}$ with i' congruent to i modulo k .

²if the index of a layer is not in the interval $[0, l]$, the layer is considered empty

Hence, there exists an integer $i \in [0, k - 1]$ such that

$$\sum_{j=-1}^{\lceil \frac{i}{k} \rceil} w(\text{OPT} \cap R_{i+kj}) \leq \frac{k+2}{k} w(\text{OPT})$$

Thus, we obtain that $w(S) \leq \frac{k+2}{k} w(\text{OPT})$. \square

9. Conclusion and Further works

In this paper, we considered a variant of the dominating set problem, called the edge monitoring problem on several classes of graphs. We also discussed the weighted version of the edge monitoring problem. In this section, we list a variety of problems for further work.

Problem 1: Study the problem on other classes of graphs: permutation graphs, strongly chordal graphs, etc.

Problem 2: Consider the following variant of the edge monitoring problem: assume that each vertex can monitor only a fixed number of edges t .

Problem 3: Consider the variant of the edge monitoring problem where the monitoring set need to be connected, namely connected edge monitoring problem.

References

- [1] G. Bagan, F. Beggas, M. Haddad, and H. Hheddouci. Edge monitoring problem on interval graphs. *Electronic Notes in Discrete Mathematics*, 54:331–336, 2016.
- [2] B. S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.
- [3] J. Baste, F. Beggas, H. Kheddouci, and I. Sau. On the parameterized complexity of the edge monitoring problem. *Information Processing Letters*, 121:39–44, 2017.
- [4] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is np-hard. *Comput. Geom.*, 9(1-2):3–24, 1998.
- [5] M. Chlebík and J. Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264–1275, 2008.
- [6] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

- [7] D. Dong, X. Liao, Y. Liu, C. Shen, and X. Wang. Edge self-monitoring for wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(3):514–527, 2011.
- [8] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.
- [9] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [10] F. Foucaud, G. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Algorithms and complexity for metric dimension and location-domination on interval and permutation graphs. In *WG*, 2015.
- [11] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem in NP complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.
- [12] M. Habib and C. Paul. A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics*, 145(2):183 – 197, 2005.
- [13] J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- [14] R. Klasing and C. Laforest. Hardness results and approximation algorithms of k-tuple domination in graphs. *Information Processing Letters*, 89(2):75–83, 2004.
- [15] T. Kloks. *Treewidth, Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer, 1994.
- [16] C. Liao and G. J. Chang. k-tuple domination in graphs. *Inf. Process. Lett.*, 87(1):45–50, 2003.
- [17] B. Neggazi, M. Haddad, V. Turau, and H. Kheddouci. A self-stabilizing algorithm for edge monitoring problem. In *Stabilization, Safety, and Security of Distributed Systems*, pages 93–105. Springer, 2014.
- [18] J. Pfaff, R. Laskar, and S. Hedetniemi. Np-completeness of total and connected domination and irredundance for bipartite graphs. Technical Report 428, Clemson University, Dept. Math. Sciences, 1982.
- [19] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Computers*, 30(2):135–140, 1981.
- [20] G. Wei, Z. Zhu, Y. Mao, and N. Xiong. A distributed node self-monitoring mechanism in wireless sensor networks. In *Information Science and Engineering (ICISE), 2010 2nd International Conference on*, pages 1684–1687. IEEE, 2010.