Parameterized Algorithm for Eternal Vertex Cover

Fedor V. Fomin[a,1], Serge Gaspers[b], Petr A. Golovach[c], Dieter Kratsch[d], Saket Saurabh[e,*]

[a]*Department of Informatics, University of Bergen, N-5020 Bergen, Norway.*
[b]*CMM – Universidad de Chile, Av. Blanco Encalada 2120, Santiago, Chile.*
[c]*School of Engineering and Computing Sciences, Durham University, South Road, DH1 3LE Durham, UK.*
[d]*Laboratoire d'Informatique Théorique et Appliquée, Université Paul Verlaine - Metz, 57045 Metz Cedex 01, France.*
[e]*The Institute of Mathematical Sciences, CIT Campus, Chennai 600 113, India.*

Abstract

In this paper we initiate the study of a "dynamic" variant of the classical VERTEX COVER problem, the ETERNAL VERTEX COVER problem introduced by Klostermeyer and Mynhardt, from the perspective of parameterized algorithms. This problem consists in placing a minimum number of guards on the vertices of a graph such that these guards can protect the graph from any sequence of attacks on its edges. In response to an attack, each guard is allowed either to stay in his vertex, or to move to a neighboring vertex. However, at least one guard has to fix the attacked edge by moving along it. The other guards may move to reconfigure and prepare for the next attack. Thus at every step the vertices occupied by guards form a vertex cover. We show that the problem admits a kernel of size $4^k(k+1) + 2k$, which shows that the problem is fixed parameter tractable when parameterized by the number of available guards k. Finally, we also provide an algorithm with running time $\mathcal{O}(2^{O(k^2)} + nm)$ for ETERNAL VERTEX COVER, where n is the number of vertices and m the number of edges of the input graph. In passing we also observe that ETERNAL VERTEX COVER is NP-hard, yet it has a polynomial time 2-approximation algorithm.

Keywords: parameterized complexity, fixed parameter tractability, vertex cover, eternal vertex cover

1. Introduction

In the classical VERTEX COVER problem we are given a graph $G = (V, E)$ and a positive integer r as input and the objective is to decide whether there exists a subset C of the vertex set V of size at most r such that it contains at least one vertex from each edge in the graph. Klostermeyer and Mynhardt [7]

---

[*]Corresponding author.
[1]Supported by the Norwegian Research Council.

introduced a dynamic variant of the VERTEX COVER problem, the ETERNAL VERTEX COVER problem. In the ETERNAL VERTEX COVER problem, the input is a graph $G = (V, E)$ on n vertices and m edges. Guards are placed on the vertices of the graph in order to protect it from an infinite sequence (which is not known to the guards in advance) of attacks on the edges of the graph. In each round, one edge $uv \in E$ is attacked, and each guard either stays on the vertex it is occupying or moves to a neighboring vertex. Moreover, the guards are bound to move in such a way that at least one guard moves from u to v or from v to u. The minimum number of guards which can protect all the edges of G is called the *eternal vertex cover number* of G and is denoted by **evcn**(G). The ETERNAL VERTEX COVER problem is to compute **evcn**(G) for a graph G, and the parameterized version of the problem is to decide whether **evcn**$(G) \le k$ for a graph G and a parameter k.

In Parameterized Complexity, one asks if for an instance of size n and a parameter k, a problem can be solved in time $f(k)n^{\mathcal{O}(1)}$ where f is an arbitrary computable function independent of n. Problems that can be solved in that time are said to be *fixed parameter tractable*, and the corresponding complexity class is called FPT. We refer to the books of Downey and Fellows [5] and Niedermeier [11] for an introduction to the area.

A lot of research has been done on the VERTEX COVER problem. As observed by many, VERTEX COVER could be considered the "*Drosophila*" of fixed-parameter algorithms [5, 6, 11]. The current fastest algorithm for VERTEX COVER runs in time $\mathcal{O}(1.2738^k + kn)$ [2]. The literature on parameterized complexity is full of variants of VERTEX COVER like CONNECTED VERTEX COVER [6, 10], CAPACITATED VERTEX COVER [4, 6], and PARTIAL VERTEX COVER [6, 8, 12]. In this paper we show that its dynamic variant, namely ETERNAL VERTEX COVER, is also fixed parameter tractable.

The paper is organized as follows. In Section 2 we give some basic definitions. and observe that the parameterized complexity of ETERNAL VERTEX COVER is the same when parameterized by the number of available guards and when parameterized by the size of a minimum vertex cover of the input graph. Section 3 deals with exact and parameterized algorithms for ETERNAL VERTEX COVER. We start with an $2^{\mathcal{O}(n)}$ time algorithm computing the eternal vertex cover number of an n-vertex graph. Then we prove FPT membership by providing a $k' = 4^k(k+1) + 2k$ kernel, which combined with the exact algorithm, leads to a time $2^{\mathcal{O}(k')} \cdot n^{\mathcal{O}(1)}$ algorithm. We then improve on this algorithm and give an algorithm with running time $\mathcal{O}(2^{O(k^2)} + nm)$. We conclude with some remarks in Section 4.

2. Preliminaries

All graphs in this paper are finite, undirected and without multiple edges and loops. For notation and terminology not defined in this paper we refer to [3]. Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is a *vertex cover* of G, if for every edge of E at least one endpoint is in S.

Studying the ETERNAL VERTEX COVER problem from the parameterized complexity point of view in Section 3, we use the following easy lemma from [7], which shows that the parameterized complexity of the problem remains the same when it is parameterized by the number of guards available to protect the graph or the size of a minimum vertex cover of the input graph. Let $\mathbf{vc}(G)$ denote the size of a minimum vertex cover of G.

Lemma 1. [7] *For any graph G, $\mathbf{vc}(G) \leq \mathbf{evcn}(G) \leq 2\mathbf{vc}(G)$.*

Let us remark that the three inequalities of Lemma 1 are tight: $\mathbf{vc}(P_2) = \mathbf{evcn}(P_2) = 1$ and $\mathbf{evcn}(P_3) = 2\mathbf{vc}(P_3) = 2$, where P_n denotes a path on n vertices.

3. FPT Algorithm for Eternal Vertex Cover

We start by observing that ETERNAL VERTEX COVER is indeed NP-hard by giving a simple reduction from VERTEX COVER. Then we obtain an exact algorithm running in time $2^{\mathcal{O}(n)}$ to find the eternal vertex cover number of a graph G on n vertices. A "trimmed variant" of this algorithm is crucially used in obtaining the faster fixed parameter tractable algorithm for ETERNAL VERTEX COVER. Finally, we give our kernel and the parameterized algorithm for ETERNAL VERTEX COVER.

Proposition 1. *It is NP-hard to decide whether k guards can protect all the edges of a graph.*

Proof: We reduce from the VERTEX COVER problem. Recall that it asks about the existence of a vertex cover of the size at most r in a graph G. Assume that $r < |V(G)| - 1$, since VERTEX COVER obviously remains NP-complete with this additional condition. We add a new universal vertex u and make it adjacent to each vertex of G. Then we add a new vertex v and the edge uv. Denote the resulting graph by H. We claim that G has a vertex cover of size at most r if and only if $k = r + 2$ guards can protect all the edges of H.

Assume that S is a vertex cover of G of size at most r. Then $r+2$ guards can use a strategy which maintains the following configuration: the guards occupy the vertices of S (one vertex is occupied by one guard), vertex u and some vertex $z \in V(G) \cup \{v\} \setminus S$. Clearly, the guards can take this position in the beginning of the game. Suppose that an edge xy is attacked. If both vertices x and y are occupied by a guard then these two guards simply exchange places by moving across xy. Otherwise, if $xy \in E(G)$, then at least one endpoint of xy, say x, is in S. One guard moves from x to y, another guard moves from u to x, and finally a guard from z moves to u. Assume now that xy is incident with u. Let $x = u$. Then one guard moves from u to y and another guard moves from z to u.

Suppose now that $r + 2$ guards can protect all edges of H. Since $r < |V(G)| - 1$, at least one guard always occupies vertex u. If no guard occupies

3

v, then after an attack on the edge uv at least one guard has to move to v, and at least two guards occupy u and v. This means that at most r guards occupy vertices of G. Clearly, the set of vertices occupied by these guards is a vertex cover of G of size at most r as neither u nor v are incident to an edge of G. $\square$

The following lemma provides a simple result on the approximability of the problem.

Lemma 2. *There is a 2-approximation algorithm for the* ETERNAL VERTEX COVER *problem with running time* $\mathcal{O}(\sqrt{n}\cdot m)$, *where n is the number of vertices and m is the number of edges of the input graph.*

Proof: Let M be a maximum matching of size k in a graph G. Then the size of a minimum vertex cover of G is at least k, and thus $\mathbf{evcn}(G) \geq k$. By placing guards on the endpoints of the edges in M, we use $2k$ guards. We claim that the guards occupying the endpoints of the edges of a maximum matching in G can always protect G. Indeed, the guards' strategy is as follows. Consider an attack on an edge xy. If both x and y are occupied by a guard, then these two guards swap their positions and the other guards do not move. Thus we are exactly in the same position as before the attack. If exactly one endpoint of xy, say x, is occupied by a guard, then there is an edge $xz \in M$, and y is not incident to an edge in M. Therefore, $M' = M \setminus \{xz\} \cup \{xy\}$ is also a matching of size k. We move a guard from x to y, and a guard from z to x. Thus the guards occupy the endpoints of the maximum matching M'.

Finally, a maximum matching in a graph can be found in time $\mathcal{O}(\sqrt{n}\cdot m)$ [9].
$\square$

3.1. Exact Algorithm for ETERNAL VERTEX COVER

In this section we give an exact algorithm running in time $2^{\mathcal{O}(n)}$ to find the eternal vertex cover number of a graph G on n vertices.

For every k, $1 \leq k \leq n$, we want to decide if k guards can protect the graph. To solve the problem we construct a configuration multi-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ from G as follows. Let $V(G) = \{v_1, \ldots, v_n\}$. A vertex $\mathcal{I} \in \mathcal{V}$ is a vector $((v_1, k_1), \ldots, (v_n, k_n))$ of pairs (v_i, k_i) where $k_i \in \{0, \ldots, k\}$ such that $\sum_{i=1}^{n} k_i = k$. It represents the configuration where k_i guards occupy vertex v_i, for every $i \in \{1, \ldots, n\}$.

Now for every edge e of G, we add an edge from vertex $\mathcal{I}$ to vertex $\mathcal{J}$ (we call it an *e-edge*) if by the attack on e, the guards can move from configuration $\mathcal{I}$ to configuration $\mathcal{J}$ in a way that at least one of the guards moves along e. Notice that $\mathcal{I} = \mathcal{J}$ is possible, i.e. $\mathcal{G}$ can have loops. Note also that if by the attack on e, the guards can move from configuration $\mathcal{I}$ to configuration $\mathcal{J}$, then the guards can move from configuration $\mathcal{J}$ to configuration $\mathcal{I}$ to counter the attack on e.

Lemma 3. *Deciding whether $(\mathcal{I}, \mathcal{J})$ is an e-edge can be done in time $n^{\mathcal{O}(1)}$.*

Proof: To decide whether there is an e-edge from $\mathcal{I}$ to $\mathcal{J}$, we solve a flow problem on an auxiliary directed graph H whose edges have capacities and costs. For the construction of H, make two copies V_1 and V_2 of the vertex set of G and construct a bipartite graph by connecting every vertex $v \in V_1$ to the vertices of V_2 from its closed neighborhood. We give cost 1 to all the edges constructed so far, except the two copies of e, which obtain cost 0. The capacities of the edges that have been constructed by now are set equal to k (or unrestricted capacity). We also add two vertices s and t, and connect s to all vertices of V_1, and all vertices of V_2 to t. Then, for every edge sv_i, $v_i \in V_1$ we put its capacity equal to the number of guards k_i located on v_i in configuration $\mathcal{I}$ and its cost equal to 0. Correspondingly, the capacity of edge ut, $u \in V_2$, is the number of guards on u in $\mathcal{J}$ and its cost is 0. Now the question whether there is an e-edge from $\mathcal{I}$ to $\mathcal{J}$ can be answered by finding a minimum cost flow from s to t in the auxiliary graph, which is solvable in polynomial (in n) time. Namely, $(\mathcal{I}, \mathcal{J})$ is an e-edge if and only if there is a $s - t$ flow in H of value k with cost at most $k - 1$ (i.e. at least one of the copies of e has non-zero flow). $\square$

We call a vertex $\mathcal{I}$ of $\mathcal{G}$ *safe* if for every edge $e \in E(G)$ there is an e-edge incident with $\mathcal{I}$ in $\mathcal{G}$. Vertex $\mathcal{I}$ is *unsafe* if there is an edge $e \in E(G)$ such that there is no e-edge incident with $\mathcal{I}$ in $\mathcal{G}$. We also say that the graph $\mathcal{G}$ is *safe* if all vertices of $\mathcal{G}$ are safe. We recursively check whether a vertex of $\mathcal{G}$ is safe or not and if it is unsafe then we remove it. We do this until the obtained graph is safe or has no vertices. Observe that we can check whether a vertex is safe or unsafe in time proportional to its degree, and check whether every vertex of $\mathcal{G}$ is safe or unsafe in time $\mathcal{O}(|\mathcal{E}|)$. Furthermore, the number of times we have to check whether the graph is safe or not is at most $|\mathcal{V}|$, as every time we delete at least one vertex from the graph $\mathcal{G}$. We arrive at the following lemma.

Lemma 4. *For a graph G, $\mathbf{evcn}(G) \leq k$ if and only if the configuration graph $\mathcal{G}$ for G and k is non-empty after recursively removing all the unsafe vertices of $\mathcal{G}$, and this property can be checked in time $\mathcal{O}(|\mathcal{E}| \cdot |\mathcal{V}|)$.*

Theorem 1. *Given a graph G on n vertices, it is possible to compute $\mathbf{evcn}(G)$ in time $64^n \cdot n^{\mathcal{O}(1)}$.*

Proof: To compute $\mathbf{evcn}(G)$, for every k from 1 to n, we compute the auxiliary graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The number of vertices in $\mathcal{G}$ is $|\mathcal{V}| \leq \binom{n+k-1}{k} \leq 2^{2n}$ and the number of edges $|\mathcal{E}| \leq |E(G)|^{\frac{|\mathcal{V}|(|\mathcal{V}|+1)}{2}}$. By Lemma 3, such a graph can be constructed in $|\mathcal{E}| \cdot n^{\mathcal{O}(1)}$ steps. By Lemma 4, given a graph $\mathcal{G}$, checking whether $\mathcal{G}$ is safe and thus $\mathbf{evcn}(G) \leq k$, can be performed in time $\mathcal{O}(|\mathcal{E}| \cdot |\mathcal{V}|) = 64^n \cdot n^{\mathcal{O}(1)}$. $\square$

3.2. An FPT Algorithm for ETERNAL VERTEX COVER

Let us start by defining the notion of *kernelization*. A parameterized problem Q is formally a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ, and an instance

of a parameterized problem is the pair $(x, k) \in \Sigma^* \times \mathbb{N}$, where k is called the parameter [5]. A kernelization algorithm for a parameterized problem $Q \subseteq \Sigma^* \times \mathbb{N}$ is an algorithm that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Sigma^* \times \mathbb{N}$ such that i) $(x, k) \in Q$ if and only if $(x', k') \in Q$, ii) there is a computable function g with the property that $|x'| + k' \leq g(k)$ for any instance (x, k). If such a kernelization algorithm exists for a parameterized problem Q, we say that Q has a kernel of size $g(k)$. It is well known (see, e.g. [5]) that a decidable parameterized problem is in FPT if and only if it admits a kernel, that is a reduction of the problem and its parameter to an instance of size bounded by a function of k. There are several techniques which allow to obtain kernels of polynomial size for VERTEX COVER [11]. Similar to the "crown reduction" approach, our kernelization algorithm for ETERNAL VERTEX COVER explores the relations to maximum matchings in a graph.

Our kernelization algorithm relies on the fact that at any step the positions occupied by the guards must form a vertex cover of size at most k of the graph G. It works as follows:

Step 1: Find a maximum matching M in G, and define S as the set of vertices incident with edges of M.

Step 2: If $|M| > k$, then return that $\mathbf{evcn}(G) > k$.

Step 3: Else, for every subset $T \subseteq S$ we define $\mathcal{S}(T)$ as the set of vertices of $I = V(G) \setminus S$ (I is an independent set) such that for every $u \in \mathcal{S}(T)$, $N(u) = T$. If any such set $\mathcal{S}(T)$ contains more than $k + 1$ vertices then we remove all the vertices except some $k + 1$ of them. Let $G(S)$ be the graph obtained after having considered all subsets $T \subseteq S$ with non-empty $\mathcal{S}(T)$.

This concludes the description of the kernelization algorithm. Next we show the correctness of the algorithm. By Lemma 2, $|M| \leq \mathbf{evcn}(G)$, and therefore, if $|M| > k$ then $\mathbf{evcn}(G) > k$. Notice also that S is a vertex cover of G. It remains to prove that the reduction described in Step 3 is correct.

Lemma 5. *Let G be a graph on n vertices and m edges. Then $\mathbf{evcn}(G) \leq k$ if and only if $\mathbf{evcn}(G(S)) \leq k$, and $G(S)$ can be constructed in time $\mathcal{O}(nm)$.*

Proof: The idea of the proof is that for a given subset $T \subseteq S$, any pair of vertices in $\mathcal{S}(T)$ is indistinguishable. Also we use the fact that if $|\mathcal{S}(T)| \geq k + 1$ then at least one vertex of $\mathcal{S}(T)$ is not occupied by guards.

Denote by $T_1, \ldots, T_r$ all subsets of S such that $|\mathcal{S}(T_j)| > k + 1$ in G. Let $U_j = \mathcal{S}(T_j)$ in G, and let $W_j = U_j \cap V(G(S))$ for $j \in \{1, \ldots, r\}$.

Suppose that $\mathbf{evcn}(G) \leq k$. Denote by $\mathcal{H}$ the graph obtained from the configuration graph $\mathcal{G}$ of G for k guards after removing all unsafe vertices. We choose a vertex w_j in each set W_j. For each vertex $\mathcal{I} \in V(\mathcal{H})$, we construct the vertex $\mathcal{J}(\mathcal{I})$ in the configuration graph for $G(S)$ in the following way. Recall that the vertex $\mathcal{I}$ is a vector $((v_1, k_1), \ldots, (v_n, k_n))$ where $\{v_1, \ldots, v_n\} = V(G)$ and k_i is the number of guards on v_i. To define $\mathcal{J}(\mathcal{I})$, we have to define the number of guards l_i in each $v_i \in V(G(S))$. For any $v_i \in V(G(S)) \setminus \{w_1, \ldots, w_r\}$,

let $l_i = k_i$. For $v_i = w_j$, let l_i be equal to the sum of the number of guards on the vertices of $\{v_i\} \cup (U_j \setminus W_j)$ in configuration $\mathcal{I}$. Denote by $\mathcal{H}'$ the subgraph of the configuration graph for $G(S)$ induced by the set of different vertices $\mathcal{J}(\mathcal{I})$ for all $\mathcal{I} \in V(\mathcal{H})$. Let $e \in E(G(S))$ and let $\mathcal{J}$ be a vertex of $\mathcal{H}'$. Suppose that $\mathcal{J} = \mathcal{J}(\mathcal{I})$ for $\mathcal{I} \in \mathcal{H}$. Since $\mathcal{I}$ is a safe vertex, there is a vertex $\mathcal{I}' \in V(\mathcal{H})$ such that $(\mathcal{I}, \mathcal{I}')$ is an e-edge. Observe that $(\mathcal{J}(\mathcal{I}), \mathcal{J}(\mathcal{I}'))$ is an e-edge in the configuration graph for $G(S)$. Hence all vertices of $\mathcal{H}'$ are safe, and therefore $\mathbf{evcn}(G(S)) \leq k$.

Assume now that $\mathbf{evcn}(G(S)) \leq k$. Denote by $\mathcal{H}'$ the graph obtained from the configuration graph of $G(S)$ for k guards after removing all unsafe vertices. For each vertex $\mathcal{J} \in \mathcal{H}'$, we construct a subset of the vertices of the configuration graph $\mathcal{G}$ of G by considering the set F of all possible collections $(f_1, \ldots, f_r)$ of mappings $f_j \colon W_j \to U_j$ for $j \in \{1, \ldots, r\}$. Let l_i be the number of guards in $v_i \in V(G(S))$ in the configuration corresponding to $\mathcal{J}$. We define the vertex $\mathcal{I} = ((v_1, k_1), \ldots, (v_n, k_n)) \in \mathcal{I}$ for mappings $(f_1, \ldots, f_r)$ in the following way. For each $v_i \in V(G) \setminus (U_1 \cup \cdots \cup U_r)$, we set $k_i = l_i$. For any $v_i \in W_j$, the number of guards in $f_j(v_i)$ is l_i. For the remaining vertices of U_j, the number of guards is equal to 0. Denote by $\mathcal{H}$ the subgraph of the configuration graph for G induced by all constructed vertices I for all possible collections of mappings from F. We claim that all vertices $\mathcal{H}$ are safe, and hence $\mathbf{evcn}(G) \leq k$.

To prove the claim, we consider an edge $e = uv \in E(G)$ and $\mathcal{I} \in V(\mathcal{H})$. It is necessary to prove that there is $\mathcal{I}' \in V(\mathcal{H})$ such that $(\mathcal{I}, \mathcal{I}')$ is an e-edge in $\mathcal{H}$. Assume that $\mathcal{I}$ is constructed from $\mathcal{J} \in \mathcal{H}'$ by the use of mappings $(f_1, \ldots, f_r)$. We consider two cases:

Case 1. $u, v \in V(G)$ but $u, v \notin (U_1 \setminus f_1(W_1)) \cup \ldots \cup (U_r \setminus f_r(W_r))$. Since J is a safe vertex, there is another vertex $\mathcal{J}'$ in $\mathcal{H}'$ such that $(\mathcal{J}, \mathcal{J}')$ is an e-edge in $\mathcal{H}'$. Let $\mathcal{I}'$ be the vertex of $\mathcal{H}$ constructed by the use of same mappings $(f_1, \ldots, f_r)$. Then $(\mathcal{I}, \mathcal{I}')$ is an e-edge in $\mathcal{H}$.

Case 2. $u \in S$ and $v \in U_j \setminus f_j(W_j)$ for some $j \in \{1, \ldots, r\}$. Since $|W_j| = k + 1$, there is a vertex $w \in W_j$ such that w is not occupied by guards in the position corresponding to J. Replace the mappings $(f_1, \ldots, f_r)$ by mappings $(f'_1, \ldots, f'_r)$ such that $f'_i = f_i$ if $i \neq j$, and

$$f'_j(x) = \begin{cases} f_j(x), & \text{if } x \neq w, \\ v, & \text{if } x = w. \end{cases}$$

It remains to observe that we can construct $\mathcal{I}$ from J by using mappings $(f'_1, \ldots, f'_r)$ instead $(f_1, \ldots, f_r)$. Because $v \in f'_j(W_j)$, this reduces to Case 1.

To prove the second claim of Lemma 5, it is sufficient to note that $|I| \leq n$ and hence there are at most n different non-empty sets $\mathcal{S}(T)$. Notice also that for two vertices $u, v \in I$, it is possible to check whether they have same neighborhood in the time proportional to their degrees. $\square$

Notice that this immediately implies the following theorem.

Theorem 2. *Given a graph G on n vertices and m edges and an integer k, we can obtain a kernel for* ETERNAL VERTEX COVER *with at most $k' = 4^k(k + 1) + 2k$ vertices in time $\mathcal{O}(nm)$.*

We can easily obtain an FPT algorithm by combining Theorems 1 and 2. This will give an FPT algorithm of running time $64^{k'}n^{\mathcal{O}(1)}$. In what follows, we improve upon this running time.

Step 1: Using the parameterized algorithm of Chen et al. [2], find a vertex cover S of size at most k of G in time $\mathcal{O}(1.2738^k + k \cdot n)$ or output NO if no such vertex cover exists.

Step 2: For every subset $T \subseteq S$, we define $\mathcal{S}(T)$ as the set of vertices of $I = V(G) \backslash S$ (I is an independent set) such that for every $u \in \mathcal{S}(T)$, $N(u) = T$. Observe that this partitions the vertices of the independent set I. If any set $\mathcal{S}(T)$ contains more than $k+1$ vertices then we remove all the vertices except some $k + 1$ of them. Let $G(S)$ be the graph obtained by this procedure.

Step 3: Check whether $\textbf{evcn}(G(S)) \leq k$ using the algorithm described in the previous subsection. If $\textbf{evcn}(G(S)) \leq k$ then return YES else return NO.

The correctness of the above described algorithm is not hard to see. For the time complexity observe that the size of S is at most k and that the size of $G(S)$ is bounded by $k' = 2^k(k + 1) + k$. Using the fact that the number of guards k is considerably smaller than k', we can improve the running time bound for our exact algorithm. The configuration graph $\mathcal{G}_S$ for $G(S)$ and k guards has at most

$$\binom{2^k(k + 1) + 2k - 1}{k} \leq 2^{\mathcal{O}(k^2)}$$

vertices. Hence Step 3 is executed in time $2^{\mathcal{O}(k^2)}$. Since Step 2 can be done in the time $\mathcal{O}(nm)$, we have the following theorem.

Theorem 3. *Let G be a graph on n vertices and m edges and k be a positive integer. It is possible to check whether $\textbf{evcn}(G) \leq k$ in time $\mathcal{O}(2^{O(k^2)} + nm)$.*

4. Concluding Remarks and Open Problems

In this paper we initiated study of the parameterized complexity of the ETERNAL VERTEX COVER problem. We showed that the problem is fixed parameter tractable by giving a kernel of size $4^k(k + 1) + 2k$. There are still many interesting questions regarding the problem which remain unanswered. Does there exist a polynomial size kernel for the problem? Or can we prove that no polynomial kernel exists for the problem, using the recently developed methodology to show non-existence of polynomial kernels [1]. Can we find a $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ algorithm for the problem? From the classical complexity perspective the challenging question is whether the ETERNAL VERTEX COVER problem is in NP?

Acknowledgments

We are grateful to the anonymous referees for their constructive suggestions and remarks.

References

[1] H. L. Bodlaender,R. G. Downey, M. R. Fellows and D. Hermelin, On Problems without Polynomial Kernels, J. of Comput. and Syst. Sci. 75(8), (2009), 423–434.

[2] J. Chen, I. A. Kanj, and G. Xia, Improved Parameterized Upper Bounds for Vertex Cover, Proceedings of MFCS 2006, Springer, LNCS 4162 (2006), 238–249.

[3] R. Diestel. *Graph Theory, Second edition*, Graduate Texts in Mathematics 173, Springer, 2000.

[4] M. Dom, D. Lokshtanov, S. Saurabh and Y. Villanger, Capacitated Domination and Covering: A Parameterized Perspective, Proceedings of IWPEC 2008, Springer, LNCS 5018 (2008), 78–90.

[5] R. G. Downey and M. R. Fellows, Parameterized Complexity, Springer-Verlag, 1999.

[6] J Guo, R. Niedermeier and S. Wernicke, Parameterized Complexity of Vertex Cover Variants, Theory Comput. Syst. 41(3) (2007), 501–520.

[7] W. F. Klostermeyer and C. M. Mynhardt, Edge Protection in Graphs, Australasian Journal of Combinatorics, 45 (2009), 235–250.

[8] J. Kneis, A. Langer and P. Rossmanith, Improved Upper Bounds for Partial Vertex Cover, Proceedings of WG 2008, Springer, LNCS 5344 (2008), 240–251.

[9] S. Micali and V. V. Vazirani, An $O(\sqrt{|V|} \cdot |E|)$ Algorithm for Finding Maximum Matching in General Graphs, in FOCS 1980, pp. 17–27.

[10] D. Mölle, S. Richter and P. Rossmanith, Enumerate and Expand: Improved Algorithms for Connected Vertex Cover and Tree Cover, Theory Comput. Syst. 43(2) (2008), 234–253.

[11] R. Niedermeier, Invitation to fixed-parameter algorithms, Oxford Lecture Series in Mathematics and Its Applications 31, Oxford University Press, USA, 2002.

[12] V. Raman and S. Saurabh, Short Cycles Make W-hard Problems Hard: FPT Algorithms for W-hard Problems in Graphs with no Short Cycles, Algorithmica 52(2) (2008), 203–225.