Discrete Applied Mathematics 200 (2016) 186–190

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

An FPT algorithm for the vertex cover P_4 problem

Jianhua Tu [a,*], Zemin Jin [b]

[a] School of Science, Beijing University of Chemical Technology, Beijing 100029, China
[b] Department of Mathematics, Zhejiang Normal University, Jinhua 321004, China

ARTICLE INFO

Article history:
Received 17 January 2015
Received in revised form 15 May 2015
Accepted 21 June 2015
Available online 15 July 2015

Keywords:
Graph algorithms
Parameterized complexity
Vertex cover P_4 problem
Iterative compression

ABSTRACT

A subset F of vertices of a graph G is called a vertex cover P_t (VCP_t) set if every path of order t in G contains at least one vertex from F. The vertex cover P_t (VCP_t) problem is to find a minimum VCP_t set in a graph. The VCP_t problem is NP-complete for any integer $t \geq 2$. In this paper, we restrict our attention to the VCP_4 problem and present an FPT algorithm with runtime $O^*(3^k)$ for the VCP_4 problem. The algorithm constructs a VCP_4 set of size at most k in a given graph G, or reports that no such VCP_4 set exists in G.

1. Introduction

A subset F of vertices of a graph G is called a vertex cover P_t (VCP_t) set if every path of order t in G contains at least one vertex from F. Denote by $\psi_t(G)$ the minimum cardinality of a VCP_t set in G. The vertex cover P_t (VCP_t) problem is to find a minimum VCP_t set in a graph. Or equivalently, the VCP_t problem is to find a minimum set F of vertices in G such that the graph $G[V \setminus F]$ has no (not necessarily induced) P_t. Clearly, the VCP_2 problem corresponds to the well-known vertex cover problem.

The concept of the VCP_t problem is motivated by real-life problems. The first motivation comes from the design of secure protocols for communication in wireless sensor networks (see [18]). The second motivation is related to controlling traffic in computer networks. The analogous motivation from the area of controlling traffic at street crossings can be found in [20].

The VCP_t problem is known to be NP-complete for any integer $t \geq 2$ [15]. For the weighted version of the VCP_t problem, Brešar et al. [4] presented linear-time algorithms for complete graphs, cycles and trees. Some bounds on $\psi_t(G)$ have been given in [2,3,13].

Li and Tu [16] proved that the VCP_4 problem is NP-complete for cubic graphs and presented a 2-approximation algorithm for the VCP_4 problem in cubic graphs. Devi et al. [7] proved that the VCP_4 problem is APX-complete for cubic graphs as well as $K_{1,4}$-free graphs and presented a 2-approximation algorithm for the VCP_4 problem in regular graphs and a 3-approximation algorithm for the VCP_4 problem in $K_{1,4}$-free graphs.

Many graphs that appear in practical applications are sparse, meaning that there are few edges (compared to the maximum number of edges possible). Hence, in a real world application it can be expected that the size k of the minimum VCP_4 set, i.e., the number of vertices in the VCP_4 set, is fairly small. This motivated the study of *parameterized algorithm* for the VCP_4 problem. Parameterized algorithm analysis is a multi-dimensional analysis of the runtime as a function of the input size and parameter(s). A parameterized problem (I, k) is *fixed-parameter tractable* (FPT) with respect to the parameter k if it

* Corresponding author.
E-mail addresses: tujh81@163.com (J. Tu), zeminjin@zjnu.cn (Z. Jin).

http://dx.doi.org/10.1016/j.dam.2015.06.032
0166-218X/

```
The compression routine (G, F)
Input: A graph G = (V, E) and a VCP_4 set F for G.
Output: A smaller VCP_4 set F' for G, if it exists, otherwise F.
for each subset Y ⊊ F
    X ← F \ Y
    G' ← G − Y
    if G[X] is a P_4-free graph, then
        X' = SV-VCP_4(G', X, V(G') \ X, |X| − 1)
        if X' ≠ "No" then return X' ∪ Y
return F
```

Fig. 1. Pseudo-code of the compression routine for the VCP_4 problem. Algorithm SV-$VCP_4(G', X, V(G') \setminus X, |X| - 1)$ solves the Disjoint Compression VCP_4 problem.

can be solved in $f(k) \cdot (|I|)^{O(1)}$ time, where f is an *arbitrary* function depending only on k, $|I|$ is the input size of the instance. Such an algorithm is referred to as an FPT algorithm for the problem (see [8,17] for more details). Thus, an FPT algorithm becomes practically efficient when the value k is small. The runtime $f(k) \cdot (|I|)^{O(1)}$ of an FPT algorithm is generally denoted as $O^*(f(k))$ by suppressing the polynomial terms.

The fixed-parameter tractability of the VCP_4 problem was first shown in [5] by a trivial FPT algorithm with runtime $O^*(4^k)$. In this paper, we present an $O^*(3^k)$ FPT algorithm for the VCP_4 problem. The general approach of our algorithm is based on the *iterative compression* method which has led to the design of FPT algorithms for many other problems [6,10–12,14,19].

2. Iterative compression for the VCP_4 problem

2.1. Preliminaries

We consider finite, simple and undirected graphs G. $V(G)$ and $E(G)$ denote its vertex set and its edge set, respectively. For $v \in V(G)$, denote by $N(v)$ the set of neighbors of v. The degree $d(v)$ is equal to $|N(v)|$. For a subset $X \subseteq V(G)$, the subgraph induced by X is denoted by $G[X]$. The difference of two sets, written $A \setminus B$ is the set of all elements of A that are not elements of B. For simplicity we will use the notation $G - w$ and $G - W$ for respectively $G[V \setminus \{w\}]$ and $G[V \setminus W]$ where $w \in V$ and $W \subset V$.

As usual, P_t denotes the path with t vertices. A triangle is a cycle with 3 vertices. For $s \geqslant 2$, the graph $K_{1,s} = (\{u, v_1, \ldots, v_s\}, \{uv_1, \ldots, uv_s\})$ is a star. The vertex u is the *center* of the star and the vertices $v_1, \ldots, v_s$ are the *leaves* of the star. *Note that according to the definition of a star, an isolated edge is not a star.* In all graph problems that follow, n denotes the number of vertices and m denotes the number of edges. For all terminologies and notations not defined here, we refer the reader to [1].

If G has no path on 4 vertices then we refer to G as a P_4-free graph, though the path mentioned here is not necessarily induced.

The parameterized version of the VCP_4 problem we study is formally defined as follows:

Input: A graph $G = (V, E)$ and a nonnegative integer k.
Task: Find a subset $F \subseteq V$ of vertices with $|F| \leqslant k$ such that each path of order 4 in G contains at least one vertex from F (the removal of all vertices in F from G results in a P_4-free graph).

Note if F is a VCP_4 set for $G = (V, E)$, then each connected component of $G[V \setminus F]$ is an isolated vertex, or an isolated edge, or a star, or a triangle.

2.2. Iterative compression framework

We present an FPT algorithm for the VCP_4 problem using the *iterative compression* method. We give a general iterative compression framework for the VCP_4 problem. First, we show how to employ the compression routine.

Iteration. We start with empty vertex subsets $V' = \emptyset$ and $F = \emptyset$; clearly, an empty set is a VCP_4 set for an empty graph. Iterating over all graph vertices, step by step we add one vertex $v \in V \setminus V'$ to both V' and F. Then F is still a VCP_4 set for $G[V']$, although possibly not a minimum one. In each step we try to find a smaller VCP_4 set for $G[V']$ by applying a *compression routine* (described below). It takes the graph $G[V']$ and a VCP_4 set F for $G[V']$, and returns a smaller VCP_4 set for $G[V']$, or proves that F is optimal (by returning a VCP_4 set of the same size). If F is optimal and $|F| > k$, then we can conclude that G does not have a VCP_4 set of size at most k. Since eventually $V' = V$, we obtain a VCP_4 set of size at most k for G once the algorithm returns F.

Compression. It remains to describe the *compression routine*. Given a graph G and a solution F for the VCP_4 problem, the compression routine finds a smaller solution for G or proves that the solution F is of minimum size. The compression routine works as follows. See Fig. 1 for the corresponding pseudo-code. Consider a smaller VCP_4 set F' as a modification of the larger