Discussiones Mathematicae
Graph Theory 39 (2019) 55–65
doi:10.7151/dmgt.2060

AN EFFICIENT POLYNOMIAL TIME APPROXIMATION SCHEME FOR THE VERTEX COVER P_3 PROBLEM ON PLANAR GRAPHS

Jianhua Tu

School of Science
Beijing University of Chemical Technology
Beijing 100029, China

e-mail: tujh81@163.com

and

Yongtang Shi [1]

Center for Combinatorics and LPMC
Nankai University
Tianjin 300071, China

e-mail: shi@nankai.edu.cn

Abstract

Given a graph $G = (V, E)$, the task in the vertex cover P_3 (VCP_3) problem is to find a minimum subset of vertices $F \subseteq V$ such that every path of order 3 in G contains at least one vertex from F. The VCP_3 problem remains NP-hard even in planar graphs and has many applications in real world. In this paper, we give a dynamic-programming algorithm to solve the VCP_3 problem on graphs of bounded branchwidth. Using the dynamic programming algorithm and the Baker's EPTAS framework for NP-hard problems, we present an efficient polynomial time approximation scheme (EPTAS) for the VCP_3 problem on planar graphs.

Keywords: combinatorial optimization, vertex cover P_3 problem, branchwidth, planar graphs, EPTAS.

2010 Mathematics Subject Classification: 05C85.

---

[1] Corresponding author.

1. Introduction

For a graph G, we denote its vertex set and edge set by $V(G)$ and $E(G)$, respectively. Unless stated otherwise, let $n := |V(G)|$ and $m := |E(G)|$. Given a graph G and an integer $t \geq 2$, a vertex subset F is called a vertex cover P_t (VCP_t) set if every path of order t in G contains at least one vertex in F. The task in the VCP_t problem is to find a minimum VCP_t set in G. Clearly, the VCP_2 problem corresponds to the well-studied vertex cover problem. Thus, the VCP_t problem can be seen as a natural generalization of the vertex cover problem. The study of the VCP_t problem is also motivated by two real-world problems. The first motivation comes from the area of controlling traffic at street crossing [22]. The second motivation is related to the design of secure protocols for communication in wireless sensor networks [6, 18].

In this paper, we restrict our attention to the VCP_3 problem. The VCP_3 problem remains NP-hard even in planar graphs with vertex degree at most 3 [21]. For the VCP_3 problem, Kardoš *et al.* [14] presented an exact algorithm with runtime $1.5171^n \cdot n^{O(1)}$ and a randomized approximation algorithm whose expected approximation ratio is 23/11. Chang *et al.* [7] gave an improved exact algorithm with runtime $1.4658^n \cdot n^{O(1)}$. Xiao *et al.* [25] gave the best exact algorithms for the VCP_3 problem which are the $O(1.4656^n)$-time polynomial space algorithm and the $O(1.3659^n)$-time exponential-space algorithm. Tu *et al.* [22, 23] presented two 2-approximation algorithms for weighted version of the VCP_3 problem using the primal-dual method and the local-ratio method. The dual problem of the VCP_3 problem is known as the maximum dissociation set problem. A vertex subset S is called a dissociation set if it induces a subgraph with vertex degree at most 1 and the maximum dissociation set problem is to find a dissociation set of maximum size. The maximum dissociation set problem has many applications in real world and has been studied extensively [4, 14, 25–27].

Parameterized complexity of the VCP_3 problem has also been studied. With the size k of the solution as the parameter, an parameterized algorithm running in time $2^k \cdot n^{O(1)}$ was given independently by Fellows *et al.* [10], Moser *et al.* [17] and Tu [20]. Using the measure & conquer approach, Wu [24] presented an improved algorithm with runtime $1.882^k \cdot n^{O(1)}$. Katrenič [15] gave further improvement so that running time reaches $1.8172^k \cdot n^{O(1)}$. The fastest parameterized algorithm is due to Chang *et al.* [8] running in $1.7964^k \cdot n^{O(1)}$ time. The connected vertex cover P_3 ($CVCP_3$) problem is a variation of the VCP_3 problem, which aims at finding a VCP_3 set F with minimum cardinality such that $G[F]$ is connected. Bodlaender *et al.* [3] presented two approaches for giving deterministic algorithms for connectivity problems that are single exponential in the treewidth.

In this paper, we consider the VCP_3 problem on planar graphs. Because of the NP-completeness of the VCP_3 problem on planar graphs [27], we design good approximation algorithms. A polynomial-time approximation scheme (PTAS) of an NP optimization problem Q is an algorithm A_Q that takes a pair (x, ε) as input, where x is an instance of Q and $\varepsilon > 0$ is a real number, and returns a feasible solution y for x such that the approximation ratio of the solution y is bounded by $1 + \varepsilon$. And, for any fixed $\varepsilon > 0$, the runtime of the algorithm A_Q is bounded by a polynomial of $|x|$. Furthermore, a PTAS is called an efficient polynomial time approximation scheme (EPTAS) if its runtime is bounded by $O(f(\varepsilon)|x|^c)$, where f is a function and c is a constant.

Baker [1] introduced a framework to obtain EPTAS for NP-complete problems on planar graphs. For an NP-complete problem, the framework is based on decomposing a planar graph into p-outerplanar subgraphs, and combing an exact algorithm for solving the problem on p-outerplanar subgraphs. Our algorithm is performed in two steps. Firstly, we present a dynamic programming algorithm for the VCP_3 problem on graphs of bounded branchwidth. Since that for p-outerplanar graphs, the branchwidth is at most $2p+1$ and a branch decomposition of width at most $2p+1$ can be obtained in $O(pn)$ time, an exact algorithm for solving the VCP_3 problem on p-outerplanar graphs can be obtained. Secondly, using Baker's approach, we give an EPTAS for the VCP_3 problem on planar graphs.

The remaining part of this paper is organized as follows. In Section 2, we give some notation and introduce the concepts of branch decomposition and branchwidth. In Section 3, we present a dynamic programming algorithm for the VCP_3 problem on graphs of bounded branchwidth. In Section 4, an EPTAS for the VCP_3 problem on planar graphs is given.

2. Preliminaries

In this paper we consider only simple, finite and undirected graphs. Given a graph $G = (V, E)$ and a vertex $v \in V$, denote by $N_G(v)$ the set of neighbors of v in G. Let $d_G(v) := |N_G(v)|$. For a subset $V' \subseteq V$, denote by $G[V']$ the subgraph induced by V'. We use $G - V'$ as an abbreviation for the induced subgraph $G[V \setminus V']$. For all terminology and notation not defined here, we refer the reader to [5].

Robertson and Seymour [19] introduced the concept of branchwidth which plays an important role in their fundamental work on graph minors. Given a graph $G = (V, E)$, a *branch decomposition* is a pair (T, τ), where T is a tree in which each non-leaf node has exactly three neighbors, and τ is a bijection from $E(G)$ to the set of leaves of T. The *order function* ω: $E(T) \to 2^{V(G)}$ of a branch decomposition maps every edge e of T to a vertex subset $\omega(e) \subseteq V(G)$ as follows. The set $\omega(e)$ consists of all vertices of $V(G)$ such that, for every vertex $v \in \omega(e)$, there exist edges $f_1, f_2 \in E(G)$ such that both f_1 and f_2 are incident with the vertex v, and the leaves $\tau(f_1)$, $\tau(f_2)$ are in different subtrees of $T - \{e\}$. The *width* of (T, τ) is equal to $\max_{e \in E(T)} |\omega(e)|$ and the *branchwidth* of G, $bw(G)$, is the minimum width over all branch decomposition of G.

For a planar graph G, a branch-decomposition of minimum width $bw(G)$ can be computed in $O(n^3)$ time [12] and $bw(G) \leq \sqrt{4.5n}$ [11].

3. The VCP_3 Problem on Graphs of Bounded Branchwidth

In this section, we present a dynamic programming algorithm to solve the VCP_3 problem on graphs G of bounded branchwidth. It is worth mentioning that our algorithm is inspired by Demaine *et al.'s* algorithm for (k, r)-centers on graphs of bounded branchwidth [9]. However, we use a different approach to define subproblems. Let G be a graph and (T, τ) be a branch decomposition of it with width at most l. Let $\omega : E(T) \to 2^{V(G)}$ be the order function of (T, τ). We choose an arbitrary edge (x, y) in T and subdivide the edge (x, y), that is to delete (x, y), add a new vertex z, and join z to x and y. Make z adjacent to a new vertex r. By choosing r as a root in the new tree $T' = T \cup \{z, r\}$, we obtain a rooted tree T' with the root r. For every edge of $e \in E(T) \cap E(T')$, we put $\omega'(e) = \omega(e)$. Also we put $\omega'((x, z)) = \omega'((z, y)) = \omega((x, y))$ and $\omega'((z, r)) = \emptyset$.

For an edge e of T' we define $E_e(V_e)$ as the set of all edges (vertices) h such that every path containing h and (z, r) in T' contains e. With such a notation, $E_{(z,r)} = E(T')$ and $V_{(z,r)} = V(T')$. We denote by $G_e = (V(G_e), E(G_e))$ the subgraph of G such that

(1) $E(G_e) = \left\{\tau^{-1}(x) | x \in V_e \text{ and } x \text{ is a leaf of } T'\right\}$, and

(2) $V(G_e)$ consists of all ends of edges of $E(G_e)$ in G.

Thus, G_e is an induced subgraph by the edge set $E(G_e)$. Note that if $e \in T'$ is a nonleaf edge and e_1, e_2 are two children of e, then $E(G_{e_1}) \cap E(G_{e_2}) = \emptyset$.

We define the subproblems. For every edge e of T', we compute partial solutions according to how a minimum VCP_3 set intersects the vertex subset $\omega'(e)$. A coloring of $\omega'(e)$ is a mapping $c : \omega'(e) \to \{1, 0_0, 0_1\}$ assigning three different colors to vertices of $\omega'(e)$. Clearly, there exist $3^{|\omega'(e)|}$ colorings of $\omega'(e)$. For a coloring c of $\omega'(e)$, denote by $A_e(c)$ the minimum size of a VCP_3 set $F \subseteq V(G_e)$ in G_e such that

- $c(v) = 1$ means that the vertex v is contained in F,
- $c(v) = 0_0$ means that the vertex v is not contained in F and is an isolated vertex in $G_e - F$,
- $c(v) = 0_1$ means that the vertex v is not contained in F and has degree at most 1 in $G_e - F$.

We put $A_e(c) = +\infty$ if no such VCP_3 set F for e and c exists. Because $\omega'((z,r)) = \emptyset$ and $G_{(z,r)} = G$, $A_{(z,r)}(\emptyset)$ is the smallest size of a VCP_3 set in G.

Compute the value of $A_e(c)$ in a bottom-up fashion. Consider, first, the values of $A_e(c)$ from leaves of T'. Let x be a leaf of T' such that $\tau^{-1}(x) = (u,v)$ and let e be the edge of T' incident with x. Then, $G_e = \{\{u,v\},\{(u,v)\}\}$. If u or v, say u, is a leaf vertex in G, then $\omega'(e) = \{v\}$ and there is always a minimum VCP_3 set in G that does not contain u. Thus,

$$A_e(c) = \begin{cases} 1, & \text{if } c(v) = 1; \\ 0, & \text{if } c(v) = 0_1; \\ +\infty, & \text{if } c(v) = 0_0. \end{cases}$$

If neither u nor v is a leaf vertex in G, then $\omega'(e) = \{u,v\}$ and

$$A_e(c) = \begin{cases} 2, & \text{if } c(u) = c(v) = 1; \\ 1, & \text{if } c(u) = 1 \text{ or } c(v) = 1, \text{ and } c(u) \neq c(v); \\ 0, & \text{if } (c(u), c(v)) = (0_1, 0_1). \\ +\infty, & \text{otherwise.} \end{cases}$$

Let e be a nonleaf edge of T' and let e_1, e_2 be the children of e. Define $X_1 = \omega'(e) - \omega'(e_2)$, $X_2 = \omega'(e) - \omega'(e_1)$, $X_3 = \omega'(e) \cap \omega'(e_1) \cap \omega'(e_2)$, and $X_4 = (\omega'(e_1) \cup \omega'(e_2)) - \omega'(e)$.

By definition of order function ω', it can never happen that a vertex belongs to exactly one of $\omega'(e), \omega'(e_1), \omega'(e_2)$. Therefore, condition $u \in X_4$ implies that $u \notin \omega'(e)$ and $u \in \omega'(e_1) \cap \omega'(e_2)$. We conclude that (see Figure 1)

$$\begin{aligned} \omega'(e) &= X_1 \cup X_2 \cup X_3, \\ \omega'(e_1) &= X_1 \cup X_3 \cup X_4, \\ \omega'(e_2) &= X_2 \cup X_3 \cup X_4. \end{aligned}$$

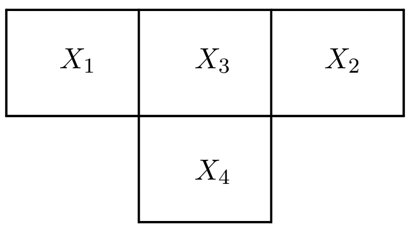

Figure 1. X_1, X_2, X_3 and X_4 are shown.

We say that colorings c_1 of $\omega'(e_1)$ and c_2 of $\omega'(e_2)$ are consistent with a coloring c of $\omega'(e)$ if

(1) for every vertex $v \in X_1$, $c_1(v) = c(v)$,

(2) for every vertex $v \in X_2$, $c_2(v) = c(v)$,

(3) for every vertex $v \in X_3$,

  (a) if $c(v) = 1$, then $c_1(v) = c_2(v) = 1$,

  (b) if $c(v) = 0_0$, then $c_1(v) = c_2(v) = 0_0$,

  (c) if $c(v) = 0_1$, then $(c_1(v), c_2(v)) \in \{(0_0, 0_1), (0_1, 0_0)\}$. By definitions of G_{e_1} and G_{e_2}, we have that $E(G_{e_1}) \cap E(G_{e_2}) = \emptyset$. Thus, if $c(v) = 0_1$, we can require that $c_1(v), c_2(v) \in \{0_0, 0_1\}$ and $c_1(v) \neq c_2(v)$.

(4) for every vertex $v \in X_4$,

  (a) either $c_1(v) = c_2(v) = 1$,

  (b) or $(c_1(v), c_2(v)) \in \{(0_0, 0_1), (0_1, 0_0)\}$.

Let $\#_1(X_3, c)$ be the number of vertices in X_3 colored by color 1 in coloring c and let $\#_1(X_4, c_1)$ be the number of vertices in X_4 colored by color 1 in coloring c_1. Since every vertex in X_3 and X_4 colored by color 1 is counted twice in $A_{e_1}(c_1) + A_{e_2}(c_2)$, the value $A_e(c)$ can be obtained from the following formula

$$A_e(c) = \min_{c_1, c_2}\{A_{e_1}(c_1) + A_{e_2}(c_2) - \#_1(X_3, c) - 2\#_1(X_4, c_1)\} \tag{1}$$

where the minimum is taken over all coloring c_1, c_2 consistent with c.

The description of the computations of the values $A_e(c)$ is complete. Let us analyze the running time of the algorithm. Clearly, the time needed to process each leaf edge of T' is constant. We consider the runtime of computing the values $A_e(c)$ of the nonleaf edge e. Note that for a coloring c of $\omega'(e)$, if a pair c_1, c_2 is consistent with c, then

(1) for every $v \in X_1 \cup X_2$,

$$(c(v), c_1(v), c_2(v)) \in \{(1, 1, 1), (0_0, 0_0, 0_0), (0_1, 0_1, 0_1)\},$$

(2) for every $v \in X_3$,

$$(c(v), c_1(v), c_2(v)) \in \{(1, 1, 1), (0_0, 0_0, 0_0), (0_1, 0_0, 0_1), (0_1, 0_1, 0_0)\},$$

(3) for every $v \in X_4$,

$$(c_1(v), c_2(v)) \in \{(1, 1), (0_1, 0_0), (0_0, 0_1)\}.$$

Let $x_i = |X_i|, 1 \leq i \leq 4$. There are exactly $3^{x_1+x_2+x_4}4^{x_3}$ triples of colorings (c, c_1, c_2) such that c_1 and c_2 are consistent with c. Thus, in order to estimate (1), the number of operations for all possible colorings of $\omega'(e)$ is $3^{x_1+x_2+x_4}4^{x_3}$. Recall that the branchwidth of G is l. The sets X_i, $1 \leq i \leq 4$, are pairwise disjoint and

$$|\omega'(e)| = x_1 + x_2 + x_3 \leq l,$$
$$|\omega'(e_1)| = x_1 + x_3 + x_4 \leq l,$$
$$|\omega'(e_2)| = x_2 + x_3 + x_4 \leq l.$$

Consider the linear function $\log_3(3^{x_1+x_2+x_4}4^{x_3}) = x_1 + x_2 + x_4 + x_3 \cdot \log_3 4$. The maximum value of the linear function subject to the above constraints is $3l/2$ (the value achieves maximum when $x_1 = x_2 = x_4 = \frac{1}{2}l, x_3 = 0$). Thus, the algorithm spends $3^{3l/2}$ time for every nonleaf edge. Since that the number of edges in T' is $O(m)$, the runtime of the dynamic programming algorithm is $O(3^{3l/2}m)$. Moreover, by bookkeeping the colorings assigned to each set $\omega'(e)$, we can construct an optimal VCP_3 set in G with the same running time. Hence, we obtain the following theorem.

Theorem 1. *For a graph G with m edges and a given branch decomposition of width at most l, there exists a dynamic programming algorithm for solving the VCP_3 problem in G with runtime $O(3^{3l/2}m)$.*

4. An EPTAS for the VCP_3 Problem on Planar Graphs

In this section, we use Baker's EPTAS framework [1] for NP-hard problems to obtain an EPTAS for the VCP_3 problem on planar graphs.

A planar graph G is called outerplanar or 1-outerplanar graph if it has a planar embedding such that all vertices of G belong to the exterior face of the embedding. For $p > 1$, a planar graph G is a p-outerplanar graph, if it has a planar embedding such that removing the vertices of G which belong to the exterior face will result in a $(p-1)$-outerplanar graph. Since a branch decomposition of a p-outerplanar graph of width at most $2p+1$ can be obtained in $O(pn)$ time [13] and for a planar graph with at least three vertices, $m \leq 3n-6$, by Theorem 1 we have

Theorem 2. *Let p be a fixed positive integer. The VCP_3 problem on p-outerplanar graphs can be solved in time $O(27^p n)$.*

For every planar graph G, there exist some p such that G is p-outerplanar. Given a planar graph G, a p-outerplanar embedding of G for which p is minimal can be found in polynomial time [2]. We define levels of vertices in a planar embedding of a planar graph G. A vertex is at level 1 if it belong to the exterior face. For $i \geq 1$, let G_i be the plane graph obtained from removing vertices at levels $1, 2, \ldots, i$ from G. Then the vertices belonging to the exterior face of G_i are at level $i+1$. A planar embedding is r-level if it has some vertices at level r and has no vertices at level $> r$. Moreover, levels of vertices can be computed in linear time [16].

Given a planar graph G with an r-level planar embedding of it, let V_1, $V_2, \ldots, V_r$ be the set of vertices at levels $1, 2, \ldots, r$, respectively.

Let k be a fixed integer. For each i, $1 \le i \le k$, let $t_i = \lceil (r-2i)/2k \rceil$. Define $U_0^i = \bigcup_{s=1}^{2i} V_s$, $U_j^i = \bigcup_{s=0}^{2k+1} V_{(j-1)\times 2k+2i-1+s}$ for $1 \le j < t_i$ and $U_{t_i}^i = \bigcup_{s=(t_i-1)\times 2k+2i-1}^{r} V_s$. For example, suppose that $r = 21$ and $k = 4$, then $1 \le i \le 4$ and when $i = 2$, $t_i = \lceil (r-2i)/2k \rceil = \lceil (21-4)/8 \rceil = 3$,

$$\begin{aligned} U_0^2 &= V_1 \cup V_2 \cup V_3 \cup V_4, \\ U_1^2 &= V_3 \cup V_4 \cup V_5 \cup \cdots \cup V_{11} \cup V_{12}, \\ U_2^2 &= V_{11} \cup V_{12} \cup V_{13} \cup \cdots \cup V_{19} \cup V_{20}, \\ U_3^2 &= V_{19} \cup V_{20} \cup V_{21}. \end{aligned}$$

For each i, $1 \le i \le k$, $G[U_0^i]$ is the subgraph induced by the vertices at levels 1 to $2i$ and is a $2i$-outerplanar subgraph, each $G[U_j^i]$ is the subgraph induced by the vertices at levels $(j-1) \times 2k + 2i - 1$ to $j \times 2k + 2i$ and is a $(2k+2)$-outerplanar subgraph for $1 \le j < t_i$, and $G[U_{t_i}^i]$ is the subgraph induced by the vertices at levels $(t_i - 1) \times 2k + 2i - 1$ to r and is a q-outerplanar subgraph, where $q = r - ((t_i - 1) \times 2k + 2i) + 2 \le 2k + 2$.

Note that $U_j^i \cap U_{j+1}^i = V_{j\times 2k+2i-1} \cup V_{j\times 2k+2i}$ for each $1 \le i \le k$, $0 \le j < t_i$. For each i, $1 \le i \le k$, let $W^i := \bigcup_{j=0}^{t_i-1} \left(U_j^i \cap U_{j+1}^i\right)$. It is easy to see that for two different i_1, i_2, $1 \le i_1, i_2 \le k$, we have $W^{i_1} \cap W^{i_2} = \emptyset$.

For fixed $\varepsilon > 0$, we design an approximation algorithm that achieves solutions at most $(1+\varepsilon)$ optimal for the VCP_3 problem on planar graphs.

Algorithm 1
Input: A planar graph G and a positive number ε.
Output: A VCP_3 set F in G of size at most $(1+\varepsilon)$ optimal.
1. Find a plane embedding of G.
2. Compute the level of every vertex and the level r of the embedding.
3. Let $V_1, V_2, \ldots, V_r$ be the set of vertices at levels $1, 2, \ldots, r$, respectively.
4. Let $k := \lceil 1/\varepsilon \rceil$.
5. for $i = 1, \ldots, k$
   5.1. $t_i = \left\lceil \frac{r-2i}{2k} \right\rceil$.
   5.2. Compute subgraphs $G[U_j^i]$ for $j = 0, 1, \ldots, t_i$.
   5.3. For every subgraph $G[U_j^i]$, find an optimal VCP_3 set F_j^i by Theorem 2.
   5.4. Let $F^i = \bigcup_{j=0}^{t_i} F_j^i$.
6. Let F be a set of $F^1, F^2, \ldots, F^k$ with the minimum cardinality.
7. Output F.

Theorem 3. *Given a fixed positive number ε, Algorithm 1 gives an $O\big(27^{2\lceil 1/\varepsilon \rceil} \cdot 1/\varepsilon \cdot n\big)$ time $(1+\varepsilon)$-approximation algorithm for the VCP_3 problem on planar graphs.*

Proof. We first show the approximation ratio of Algorithm 1. Let $k := \lceil 1/\varepsilon \rceil$. For each i, $1 \le i \le k$, an optimal VCP_3 set F^i_j in each subgraph $G[U^i_j]$ can be obtained by Theorem 2. Recall that $U^i_j \cap U^i_{j+1} = V_{j\times 2k+2i-1} \cup V_{j\times 2k+2i}$ for each $0 \le j < t_i$, in other words, two consecutive U^i_j overlap by two levels. Thus it is easy to see that $F^i = \bigcup_{j=0}^{t_i} F^i_j$ is a VCP_3 set of G. Because F is a set of $F^1, F^2, \ldots, F^k$ with the minimum cardinality, the output F of Algorithm 1 is a VCP_3 set of G.

For each i, $1 \le i \le k$, let $W^i := \bigcup_{j=0}^{t_i-1} \big(U^i_j \cap U^i_{j+1}\big)$. Recall that for two different i_1, i_2, $1 \le i_1, i_2 \le k$, we have $W^{i_1} \cap W^{i_2} = \emptyset$.

Let $\widetilde{F}$ be a minimum VCP_3 set of G. There exist some h, $1 \le h \le k$, such that $\big|\widetilde{F} \cap W^h\big| \le \big|\widetilde{F}\big|/k$. Let $\widetilde{F}^h_j := \widetilde{F} \cap U^h_j$ for each $0 \le j \le t_h$. Thus, $\widetilde{F}^h_j$ is a VCP_3 set of $G\big[U^h_j\big]$ and $\big|F^h_j\big| \le \big|\widetilde{F}^h_j\big|$ for each $0 \le j \le t_h$. Hence,

$$\sum_{j=0}^{t_h} \big|\widetilde{F}^h_j\big| = \sum_{j=0}^{t_h} \big|\widetilde{F} \cap U^h_j\big| = \big|\widetilde{F}\big| + \big|\widetilde{F} \cap W^h\big| \le (1 + 1/k)\big|\widetilde{F}\big|,$$

and

$$|F| \le \big|F^h\big| \le \sum_{j=0}^{t_h} \big|F^h_j\big| \le \sum_{j=0}^{t_h} \big|\widetilde{F}^h_j\big| \le (1 + 1/k)\big|\widetilde{F}\big| \le (1 + \varepsilon)\big|\widetilde{F}\big|.$$

Thus, the solution produced by Algorithm 1 has the approximation ratio $(1 + \varepsilon)$ for the VCP_3 problem on planar graphs.

Now, we analyze the running time of Algorithm 1. Given a planar graph G, a planar embedding of G and the levels of vertices can be computed in linear time [16]. Thus, Step 1, Step 2 and Step 3 can be computed in linear time. By Theorem 2, Step 5 can be computed in $O(27^{2k}kn)$ time. Thus, the running time of Algorithm 1 is $O\big(27^{2\lceil 1/\varepsilon \rceil} \cdot 1/\varepsilon \cdot n\big)$.

The proof is completed. ■

Acknowledgment

We thank the referees for their helpful comments and suggestions.

J. Tu was supported by the National Natural Science Foundation of China (No. 11201021), BUCT Fund for Disciplines Construction and Development (Project No. 1524). Y. Shi was partially supported by the Natural Science Foundation of Tianjin (No. 17JCQNJC00300) and the National Natural Science Foundation of China.

References

[1] B.S. Baker, *Approximation algorithms for NP-complete problems on planar graphs*, J. ACM **41** (1994) 153–180.
doi:10.1145/174644.174650

[2] F. Bienstock and C.L. Monma, *On the complexity of embedding planar graphs to minimize centain distance measure*, Algorithmica **5** (1990) 93–109. doi:10.1007/BF01840379

[3] H.L. Bodlaender, M. Cygan, S. Kratsch and J. Nederlof, *Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth*, Inform. and Comput. **243** (2015) 86–111. doi:10.1016/j.ic.2014.12.008

[4] R. Boliac, K. Cameron and V.V. Lozin, *On computing the dissociation number and the induced matching number of bipartite graphs*, Ars Combin. **72** (2004) 241–253.

[5] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications (Macmillan/Elsevier, London/New York, 1976).

[6] B. Brešar, F. Kardoš, J. Katrenič and G. Semanišin, *Minimum k-path vertex cover*, Discrete Appl. Math. **159** (2011) 1189–1195. doi:10.1016/j.dam.2011.04.008

[7] M.S. Chang, L.H. Chen, L.J. Hung, Y.Z. Liu, P. Rossmanith and S. Sikdar, *An $O^*(1.4658^n)$-time exact algorithm for the maximum bounded-degree-1 set problem*, in: Proceedings of the 31st Workshop on Combinatorial Mathematics and Computation Theory (2014) 9–18.

[8] M.S. Chang, L.H. Chen, L.J. Hung, P. Rossmanith and P.C. Su, *Fixed-parameter algorithms for vertex cover P_3*, Discrete Optim. **19** (2016) 12–22. doi:10.1016/j.disopt.2015.11.003

[9] E.D. Demaine, F.V. Fomin, M. Hajiaghayi and D.M. Thilikos, *Fixed-parameter algorithms for (k,r)-center in planar graphs and map graphs*, ACM Trans. Algorithms **1** (2005) 33–47.

[10] M.R. Fellows, J. Guo, H. Moser and R. Niedermeier, *A complexity dichotomy for finding disjoint solutions of vertex deletion problems*, ACM Trans. Comput. Theory **2** (2011) #5.

[11] F.V. Fomin and D.M. Thilikos, *New upper bounds on the decomposability of planar graphs*, J. Graph Theory **51** (2006) 53–81. doi:10.1002/jgt.20121

[12] Q. Gu and H. Tamaki, *Optimal branch-decomposition of planar graphs in $O(n^3)$ time*, ACM Trans. Algorithms **4** (2008) #30.

[13] O. Hjortas, Branch decompositions of k-outerplanar graphs (Master's Thesis, University of Bergen, Department of Informatics, 2005).

[14] F. Kardoš, J. Katrenič and I. Schiermeyer, *On computing the minimum 3-path vertex cover and dissociation number of graphs*, Theoret. Comput. Sci. **412** (2011) 7009–7017. doi:10.1016/j.tcs.2011.09.009

[15] J. Katrenič, *A faster FPT algorithm for 3-path vertex cover*, Inform. Process. Lett. **116** (2016) 273–278. doi:10.1016/j.ipl.2015.12.002

[16] R.J. Lipton and R.E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Appl. Math. **36** (1979) 177–189.
doi:10.1137/0136016

[17] H. Moser, R. Niedermeier and M. Sorge, *Exact combinatorial algorithms and experiments for finding maximum k-plexes*, J. Comb. Optim. **24** (2012) 347–373.
doi:10.1007/s10878-011-9391-5

[18] M. Novotný, *Design and analysis of a generalized canvas protocol*, in: Proceedings of WISTP 2010, Lecture Notes in Comput. Sci. **6033** (2010) 106–121.
doi:10.1007/978-3-642-12368-9_8

[19] N. Robertson and P.D. Seymour, *Graph minors, X. obstructions to tree-decomposition*, J. Combin. Theory Ser. B **52** (1991) 153–190.
doi:10.1016/0095-8956(91)90061-N

[20] J.H. Tu, *A fixed-parameter algorithm for the vertex cover P_3 problem*, Inform. Process. Lett. **115** (2015) 96–99.
doi:10.1016/j.ipl.2014.06.018

[21] J.H. Tu and F.M. Yang, *The vertex cover P_3 problem in cubic graphs*, Inform. Process. Lett. **113** (2013) 481–485.
doi:10.1016/j.ipl.2013.04.002

[22] J.H. Tu and W.L. Zhou, *A primal-dual approximation algorithm for the vertex cover P_3 problem*, Theoret. Comput. Sci. **412** (2011) 7044–7048.
doi:10.1016/j.tcs.2011.09.013

[23] J.H. Tu and W.L. Zhou, *A factor* 2 *approximation algorithm for the vertex cover P_3 problem*, Inform. Process. Lett. **111** (2011) 683–686.
doi:10.1016/j.ipl.2011.04.009

[24] B.Y. Wu, *A measure and conquer approach for the parameterized bounded degree-one vertex deletion*, COCOON (2015) 469–480.
doi:10.1007/978-3-319-21398-9_37

[25] M.Y. Xiao and S.W. Kou, *Exact algorithms for the maximum dissociation set and minimum* 3*-path vertex cover problems*, Theoret. Comput. Sci. **657** (2017) 86–97.
doi:10.1016/j.tcs.2016.04.043

[26] M.Y. Xiao and H. Nagamochi, *Complexity and kernels for bipartition into degree-bounded induced graphs*, Theoret. Comput. Sci. **659** (2017) 72–82.
doi:10.1016/j.tcs.2016.11.011

[27] M. Yannakakis, *Node-deletion problems on bipartite graphs*, SIAM J. Comput. **10** (1981) 310–327.
doi:10.1137/0210022

Received 10 November 2016
Revised 8 May 2017
Accepted 8 May 2017