Discrete Applied Mathematics 287 (2020) 77–84

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

Approximation algorithm for minimum connected 3-path vertex cover

Pengcheng Liu [a], Zhao Zhang [a,*], Xianyue Li [b], Weili Wu [c]

[a] College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, Zhejiang, 321004, People's Republic of China
[b] School of Mathematics and Statistics, Lanzhou University, Lanzhou, Gansu, 730000, People's Republic of China
[c] Department of Computer Science, University of Texas at Dallas, Richardson, TX, 75080, USA

ARTICLE INFO

Article history:
Received 5 February 2019
Received in revised form 30 July 2020
Accepted 5 August 2020
Available online 22 August 2020

Keywords:
Connected k-path vertex cover
Approximation algorithm

ABSTRACT

A vertex subset S of a given graph $G = (V, E)$ is called a *connected k-path vertex cover* ($CVCP_k$) if every k-path of G contains at least one vertex from S, and the subgraph of G induced by S is connected. This concept has its background in the field of security and supervisory and the computation of a minimum $CVCP_k$ is NP-hard. In this paper, we give a $(2\alpha + 1/2)$-approximation algorithm for $MinCVCP_3$, where α is the performance ratio of an algorithm for $MinVCP_3$.

1. Introduction

Nowadays, wireless sensor networks (WSNs) have been applied widely, including battlefield monitoring, traffic control, disaster detection, and home automation, etc. Those applications bring several important research issues into studies, such as coverage, connectivity, and security. In a WSN, a sensor has limited capabilities and is vulnerable to be captured. Under such a setting, traditional security techniques such as key establishment and authentication cannot be used directly. As pointed out in [22] "design of WSN security protocols has become a challenge in the computer security research field".

Security concerns include several important properties, one of which is data integrity, which guarantees accuracy and consistency of data during the working period of a WSN. There are several protocols and models to ensure data integrity [29,30], one of which is the *Generalized Canvas Scheme* proposed by Novotný in [22]. This protocol guarantees data integrality by requiring that every path on k vertices contains a protected vertex.

In graph theoretical language, the above model is described as follows. Suppose $G = (V, E)$ is a graph on vertex set V and edge set E. A *k-path* is a path on k vertices. A vertex subset S is a *k-path vertex cover* (VCP_k) if every k-path of G contains at least one vertex from S. Since protected vertices cost more, it is desirable that the number of protected vertices is as small as possible. Thus we have the Minimum VCP_k problem ($MinVCP_k$). In particular, $MinVCP_2$ is exactly the well-known minimum vertex cover problem (MinVC). $MinVCP_k$ is also known as the *minimum k-path transversal problem* in some other literature [17].

Another application of $MinVCP_k$ is to monitor message flows in a WSN. Suppose every message which continuously passes k vertices should be monitored at least once, then we again have the $MinVCP_k$ problem. In an application of monitoring, connectivity is an important factor to be considered: in order that information collected by monitors can be shared with each other in a fast way, it is desirable that the monitors form a connected group. Then we have the *minimum connected k-path vertex cover* problem ($MinCVCP_k$), the goal of which is to find a minimum vertex subset S of graph G such that S is VCP_k of G and the subgraph of G induced by S, denoted as $G[S]$, is connected.

In this paper, we study approximation algorithm for $MinCVC_3$.

* Corresponding author.
E-mail address: hxhzz@sina.com (Z. Zhang).

https://doi.org/10.1016/j.dam.2020.08.008
0166-218X/

1.1. Related works

The MinVCP$_k$ problem is a special case of the *minimum vertex deletion problem* studied in [10,16,18], the goal of which is to select a minimum vertex set whose deletion results in a graph satisfying some specific property. If this property is that there is no k-path in the remaining graph, then it is exactly MinVCP$_k$. Notice that if S is a VCP$_3$ of G, then $V(G)\setminus S$ is a *dissociation set* of G, which is defined to be a set of vertices inducing a subgraph of maximum degree at most 1. The complement problem of MinVCP$_3$ is the maximum dissociation set problem studied in [5,23,31,32].

The minimum cardinality of a *VCP$_k$* is denoted as ψ_k. There are a lot of studies on upper and lower bounds of ψ_k from a graph theoretical point of view (see for example [2,12]). Since this paper is on the algorithmic aspect of MinVCP$_k$, we mainly focus on related algorithmic results in the following.

Brešar et al. [2] proved that MinVCP$_k$ is NP-hard for any $k\geq 2$. The proof is done by a performance preserving reduction from the minimum vertex cover problem. Combining this result with the non-approximability of MinVC in [15], it is highly likely that MinVCP$_k$ does not admit an approximation factor strictly smaller than $\sqrt{2}$. Boliac et al. [1] proved that it is NP-hard even in bipartite graphs that are C_4-free and have maximum degree 3.

For some special graph classes including complete graphs, cycles, and trees, Brešar et al. [4] showed that MinVCP$_k$ can be solved in polynomial-time even for its weighted version (denoted as MinWVCP$_k$), in which every vertex has a weight and the goal is to choose a VCP$_k$ with the minimum weight. Recently, Brešar et al. [3] presented a linear time algorithm for the MinVCP$_3$ problem on the class of P_4-tidy graphs which widely generalizes the class of cographs.

Starting from the $O(1.5171^n)$-time algorithm for MinVCP$_3$ by Kardoš et al. [13], there are a lot of studies on exact algorithms and FPT algorithms for MinVCP$_3$ [14,25,31]. Currently, the best known running time of an FPT for MinVCP$_3$ is $O^*(1.7964^K)$ in polynomial space and $O^*(1.7485^K)$ in exponential space [7], where K is the size of an optimal VCP$_3$.

Because of the NP-hardness of MinVCP$_k$, another trend is to design approximation algorithms for the problem. By noticing that MinVCP$_k$ is a special case of the minimum set cover problem (MinSC), it admits k-approximation [35]. Improvement on the ratio needs further exploration of graph structures. Kardoš et al. [13] presented a randomized algorithm for MinVCP$_3$ with expected performance ratio $\frac{23}{11}$. Tu et al. gave two 2-approximation algorithms for MinWVCP$_3$ using local ratio method [27] and primal–dual method [28], respectively. On cubic graphs, the ratio for MinVCP$_3$ can be improved to 1.57 [26] and the ratio for MinVCP$_4$ can be improved to 2 [19].

There are very few works on MinVCP$_k$ for general k. Considering geometry, Zhang et al. [34] presented a PTAS (that is, a $(1+\varepsilon)$-approximation) for MinVCP$_k$ on a ball graph (which is a widely adopted model for a heterogeneous wireless sensor network) under the assumption that the heterogeneity (which is the ratio of the maximum radius over the minimum radius of balls) has a constant upper bound. A breakthrough was made very recently by Lee in [17], who presented an $O(\log k)$-approximation algorithm with running time $2^{O(k^3\log k)}n^2\log n+n^{O(1)}$. Notice that when k is a constant, the algorithm runs in polynomial-time. Recently, Zhang et al. [33] obtained a $\frac{\lfloor\frac{d}{2}\rfloor(2d-k+2)}{(\lfloor\frac{d}{2}\rfloor+1)(d-k+2)}$-approximation for MinVCP$_k$ on d-regular graphs with $1\leq k-2<d$.

Liu et al. [21] were the first to study MinVCP$_k$ under a requirement of connectivity. They gave a PTAS for MinCVCP$_k$ on unit disk graphs (a unit disk graph is a 2-dimensional ball graph in which all balls have the same radius). A simplified approach which also yields a PTAS for MinCVCP$_k$ but runs faster was given in [8]. Notice that a basis for Liu's algorithm in [21] is a k^2-approximation algorithm for MinCVCP$_k$ on a general graph. Li et al. [20] improved the ratio to k on a graph with girth (the length of a shortest cycle) at least k. Recently, Fujito [11] removed the girth requirement, showing that the MinCVCP$_k$ problem on a general graph admits a k-approximation. The above studies are on the cardinality version. Considering weight, Ran et al. [24] presented a greedy algorithm for the minimum weight CVCP$_3$ problem in a general graph, achieving approximation ratio $\ln\Delta+4+\ln 2$, where Δ is the maximum degree of the graph. This ratio is tight under the assumption that $P\neq NP$.

1.2. Our contributions

In this paper, we present a $(2\alpha+\frac{1}{2})$-approximation algorithm for MinCVCP$_3$, where α is the performance ratio of an algorithm for MinVCP$_3$.

The main contribution of this paper is on the connecting part. The idea of our algorithm is as follows. Suppose S is a VCP$_3$ of graph G with $G[S]$ having $l\geq 2$ connected components. Since S is a VCP$_3$, adding at most two vertices can merge at least two connected components of $G[S]$. Hence adding at most $2(l-1)\leq 2(|S|-1)$ vertices will result in a CVCP$_3$ whose size is at most $3|S|-2$. So, if MinVCP$_3$ has an α-approximation, then a 3α-approximation for MinCVCP$_3$ can be easily obtained. To obtain a better result, consider the efficiency of mergence per vertex, that is, averagely speaking, how many connected components can be merged by adding one vertex. In a worst case, adding two vertices can merge two connected components, the efficiency of which is 1. The efficiency will be higher if adding two vertices can merge three or more connected components. So, our algorithm will execute such more efficient operations as long as possible. When such more efficient operations cannot go on, the graph has a special structure, which enables us to find a controllable number of vertices to connect the remaining connected components. Such an idea is inspired by [9], in which a 5/3-approximation algorithm was obtained for the minimum connected vertex cover problem (MinCVC) on those classes of graphs for which MinVC is polynomial-time solvable. Besides such an idea, due to the much more complicated structure of a CVCP$_3$ than CVC, more manipulations and more ideas are needed both for the algorithm and for the analysis.

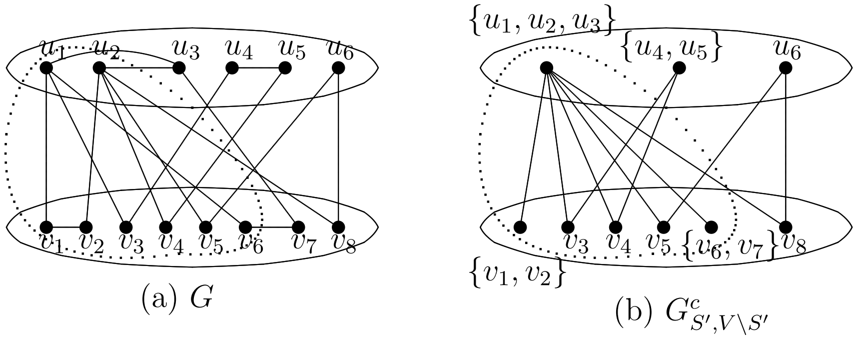

Fig. 1. An illustration for the effect of contracting connected components.

The following part is organized as follows. In Section 2, we introduce some basic notations and some preliminary results which will be used in our paper. In Section 3, a $(2\alpha + 1/2)$-approximation algorithm is presented for MinCVCP$_3$, where α is the approximation factor of an algorithm for MinVCP$_3$. The paper is concluded in Section 4 with some discussions on further study.

2. Preliminaries

All graphs considered are undirected, simple and without loops. Our algorithm needs the operation of connected contraction defined as follows.

Definition 1. For a vertex set $A \subseteq V$ of a graph $G = (V, E)$, the *contraction of G with respect to A* is the simple graph G_A which is obtained from G by identifying all vertices in A as a new vertex v_A and replacing those edges with the form of uv with $u \notin A$ and $v \in A$ by an edge uv_A, removing redundant parallel edges. We also say that G_A is obtained from G by contracting vertex set A. Suppose V' is a vertex subset of V and $G[V']$ has q connected components $G_1, \ldots, G_q$. Let $A_i = V(G_i)$. The *connected contraction of G following V'* is the graph $G^c_{V'}$ obtained from G by contracting $A_1, \ldots, A_q$.

Suppose S is a VCP$_3$ of a connected graph G. Then every connected component of $G[V \setminus S]$ is either an isolated vertex or an isolated edge. Let $G^c_{S,V\setminus S} = (G^c_S)^c_{V\setminus S}$ be the graph obtained from G by first contracting connected components of $G[S]$ and then contracting connected components of $G[V \setminus S]$. Then $G^c_{S,V\setminus S}$ is a bipartite graph with bipartition (L_S, R_S), where L_S consists of those contracted vertices of $G[S]$ and R_S consists of those contracted vertices of $G[V \setminus S]$. For each contracted vertex $s \in V(G^c_{S,V\setminus S})$, denote by A_s the set of vertices of $V(G)$ which are contracted onto s. As to those vertices in G, we call them *normal vertices*. For example, starting from graph G in Fig. 1(a), let S be the set of vertices in the upper ellipse, after identifying $\{u_1, u_2, u_3\}$ and $\{u_4, u_5\}$ which are nontrivial connected components of $G[S]$, and identifying $\{v_1, v_2\}$ and $\{v_6, v_7\}$ which are nontrivial connected components of $G[V \setminus S]$, the resulting graph $G^c_{S,V\setminus S}$ is depicted in Fig. 1(b). The dotted lines will be explained later.

In the following, we use $d_G(u)$ to denote the degree of vertex u in graph G.

Lemma 2. *Suppose S is a VCP$_3$ of G (not necessarily minimum) such that $G[S]$ is not connected and $d_{G^c_{S,V\setminus S}}(r) \le 2$ holds for every vertex $r \in R_S$. We can change S into vertex set S' in $O(n^2)$ time (where n is the number of vertices) such that*
(i) S' is also a VCP$_3$ of G (feasibility condition),
(ii) $|S'| \le |S|$ (S-monotonicity condition),
(iii) $|L_{S'}| \le |L_S|$ (L-monotonicity condition),
(iv) $d_{G^c_{S',V\setminus S'}}(r) \le 2$ for every vertex $r \in R_{S'}$ (R-degree condition), and
(v) $d_{G^c_{S',V\setminus S'}}(l) \ge 2$ holds for any vertex $l \in L_{S'}$ if $G[S']$ is not already connected (L-degree condition).

Proof. To prove the lemma, we are to show that as long as L_S has a vertex of degree one in $G^c_{S,V\setminus S}$, then we can construct a vertex set S' satisfying conditions (i) to (iv) such that

the number of degree one vertices in $L_{S'}$ is strictly smaller than the number of degree one vertices in L_S. (1)

Then iteratively using such an operation, we will eventually obtain a vertex set S' satisfying all conditions (i) to (v).

Suppose $l \in L_S$ has degree one in $G^c_{S,V\setminus S}$. Let $r \in R_S$ be the unique neighbor of l in $G^c_{S,V\setminus S}$. Recall that $G[A_r]$ is either an isolated vertex or an isolated edge in $G[V \setminus S]$.

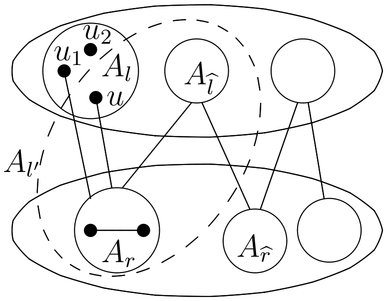

Fig. 2. An illustration for the proof of adjustment in Lemma 2 Case 1. The dashed curve indicates the new connected component $G[A_{l'}]$ of $G[S']$.

Case 1. $G[A_r]$ is an edge.

First, assume $|A_l| \geq 3$. For an illustration of the proof, the readers may refer to Fig. 2.

Let u be a vertex of A_l which is adjacent with A_r. Since $G[A_l]$ is connected, we can find two vertices $u_1, u_2 \neq u$ such that $G[A_l \setminus \{u_1, u_2\}]$ is connected. In fact, vertices u_1 and u_2 can be found as the last two vertices visited by a BFS starting from u. Let $S' = (S \setminus \{u_1, u_2\}) \cup A_r$. Then S' is a VCP$_3$ of G with $|S'| = |S|$ (conditions (i) and (ii) hold), subgraph $G[(A_l \setminus \{u_1, u_2\}) \cup A_r]$ is connected, and $\{u_1, u_2\}$ form one connected component (in the case when u_1u_2 is an edge in G) or two connected components (in the case when u_1 and u_2 are not adjacent in G) of $G[V \setminus S']$ which are only adjacent with the connected component of $G[S']$ containing $(A_l \setminus \{u_1, u_2\}) \cup A_r$.

Since $G[S]$ is not connected, we have $|L_S| \geq 2$. By the connectedness of G and the assumption that $d_{G^c_{S,V\setminus S}}(r) \leq 2$, we see that r has degree exactly 2 in $G^c_{S,V\setminus S}$. Let $\widehat{l}$ be the other neighbor of r in $G^c_{S,V\setminus S}$. Then vertex set $A_{l'} = (A_l \setminus \{u_1, u_2\}) \cup A_r \cup A_{\widehat{l}}$ induces a connected component of $G[S']$ (denote by l' the vertex in $L_{S'}$ corresponding to this connected component). Notice that all other connected components of $G[S]$ remain the same. Hence $|L_{S'}| = |L_S| - 1$, condition (iii) holds.

Notice that except for $G[A_r]$, all other connected components of $G[V \setminus S]$ are still connected components of $G[V \setminus S']$ with the same adjacency relation, and thus their corresponding vertices have the same degrees in $G^c_{S,V\setminus S}$ and $G^c_{S',V\setminus S'}$ (which are not greater than 2). Furthermore, $G[V \setminus S']$ has one or two more connected components (namely those connected components formed by u_1, u_2). Since such an extra connected component is only adjacent with $G[A_{l'}]$, its corresponding vertex has degree one in $G^c_{S',V\setminus S'}$. Condition (iv) is proved.

Finally, if $G[S']$ is not connected, then $A_{\widehat{l}}$ is adjacent with some connected component of $G[V \setminus S]$ different from $G[A_r]$, which is also a connected component of $G[V \setminus S']$. Adding the one or two connected components formed by u_1 and u_2, vertex l' has degree at least 2 in $G^c_{S',V\setminus S'}$. As to every other connected component of $G[S']$, its corresponding vertex has the same degree in $G^c_{S,V\setminus S}$ and $G^c_{S',V\setminus S'}$. So, condition (1) holds.

The case when $|A_l| = 2$ can be considered similarly by letting $S' = (S \setminus A_l) \cup A_r$. The case when $|A_l| = 1$ can be considered similarly by letting $S' = (S \setminus \{u\}) \cup \{v\}$, where the unique vertex in A_l is u, $A_r = \{v, \hat{v}\}$, and v is adjacent with some other vertex in L_S (such v exists since $G[S]$ is not connected but G is connected).

Case 2. $|A_r| = 1$.

Let v be the unique vertex in A_r and let u be a vertex in A_l such that $G[A_l \setminus \{u\}]$ is connected or empty. Let $S' = S \setminus \{u\}$. Since all neighbors of v are in S and all neighbors of u are in $S \cup \{v\}$, vertex set $\{u, v\}$ induces either two isolated vertices of $G - S'$ (if u and v are not adjacent) or an edge of $G - S'$ (if uv is an edge of G), which implies that S' is a VCP$_3$ of G with $|S'| = |S| - 1$. If vertex set $A_l \setminus \{u\} = \emptyset$, then one connected component of $G[S]$ whose corresponding vertex has degree one in $G^c_{S,V\setminus S}$ (namely the connected component $G[A_l] = \{u\}$) vanishes. In this case, all conditions (i) to (iv) and property (1) are satisfied. If $A_l \setminus \{u\} \neq \emptyset$, then by the choice of u, $G[A_l \setminus \{u\}]$ is a connected component of $G[S']$, which corresponds to a vertex l' of $G^c_{S',V\setminus S'}$. In this case, $|L_{S'}| = |L_S|$. If u and v are not adjacent, then l' has degree 2 in $G^c_{S',V\setminus S'}$ (since $\{u\}$ and $\{v\}$ are the two connected components of $G[V \setminus S']$ adjacent with $A_{l'}$). We also have conditions (i) to (iv) and property (1). If uv is an edge, then we arrive at Case 1.

Notice that the above switch operations lead to $|L_{S'}| < |L_S|$ in Case 1 and $|S'| < |S|$ in Case 2. Because $|S| + |L_S| = O(n)$, in at most $O(n)$ iterations, we could obtain a VCP$_3$ S' satisfying all conditions (i) to (v). Each iteration can be done in time $O(n)$. Hence a desired set S' can be found in time $O(n^2)$. □

Besides those operations in the proof of Lemma 2, we will also need the following operation to change a VCP$_3$ set S into another VCP$_3$ set S' (see Fig. 3).

Operation A. Suppose S is a VCP$_3$ of G. If $|L_S| \geq 2$ and there exists a vertex $u \in S$ which is an isolated vertex in $G[S]$ with $d_G(u) = 2$, and there is a neighbor of u, say w, such that $S' = (S \setminus \{u\}) \cup \{w\}$ is still a VCP$_3$ of G and the number of connected components of $G[S']$ is strictly smaller than the number of connected components of $G[S]$, then replace S by S'.

Remark 3. Notice that if S is a VCP$_3$ of G satisfying the R- and L-degree conditions of Lemma 2, then the set S' obtained through Operation A satisfies all those conditions of Lemma 2.

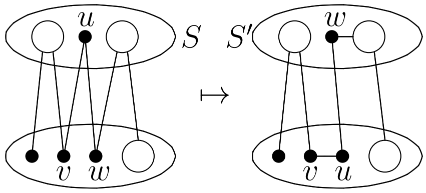

Fig. 3. An illustration of Operation A.

The next lemma is implied in a result due to Escoffier et al. (Lemma 9 of [9]).

Lemma 4. *Suppose G is a connected bipartite graph on at least 3 vertices with bipartition (L, R) in which $d_G(r) \leq 2$ holds for any $r \in R$ and $d_G(l) \geq 2$ holds for any $l \in L$. Then the cardinality of a minimum CVC of G is $2|L| - 1$ and there exists a minimum CVC containing L. Furthermore, such a minimum CVC can be found in polynomial time.*

Remark 5. The following is the method implied in the proof of Lemma 9 of [9] to find a minimum CVC of graph G which satisfies those conditions of Lemma 4. Construct an auxiliary graph H on vertex set L. Two vertices $l_1, l_2 \in L$ are adjacent in H if and only if l_1 and l_2 have a common neighbor in G. Let T be a spanning tree of H. For each edge $e \in E(T)$, let r_e be a common neighbor of the two ends of e in G (although the two ends of e may have many common neighbors, it suffices to pick only one of them to serve as r_e). Then, $L \cup \{r_e : e \in E(T)\}$ is a minimum CVC. As a consequence of this method, it can be seen that no vertex of degree one in G can belong to the minimum CVC.

3. Approximation algorithm

Our algorithm for MinCVCP$_3$ is described in Algorithm 1. An optimal CVCP$_3$ with cardinality one is a vertex, and an optimal CVCP$_3$ of cardinality two is an edge. After dealing with these two special cases in the first two lines of the algorithm, the remaining part is devoted to the case when an optimal CVCP$_3$ has at least three vertices.

Notice that a VCP$_3$ S is a CVCP$_3$ of G if and only if $|L_S| = 1$. After obtaining a VCP$_3$ S by an α-approximation algorithm, while $|L_S| \geq 2$, we add more vertices to connect S. In the while loop, as long as there exists a vertex $r \in R_S$ with degree at least 3 in $G^c_{S,V\setminus S}$ (that is, adding A_r will merge at least three connected components of $G[S]$), then add A_r into S. When the algorithm jumps out of the while loop, if we still have $|L_S| \geq 2$, then every vertex $r \in R_S$ has degree at most 2 in $G^c_{S,V\setminus S}$. After changing S into S' satisfying conditions (i) to (v) of Lemma 2, Operation A is applied recursively until it cannot be applied any more. By Remark 3, the resulting graph $G^c_{S',V\setminus S'}$ satisfies the conditions of Lemma 4. Hence a minimum CVC of $G^c_{S',V\setminus S'}$ can be found in polynomial time which contains $L_{S'}$. By expanding all those contracted vertices, we obtain a CVCP$_3$ of G.

Algorithm 1 Approximation Algorithm for CVCP$_3$.

Input: A connected graph $G = (V, E)$.
Output: A vertex set S_A which is a CVCP$_3$ of G.

```
If there is a vertex u such that G − u is P3-free, then output S_A = {u}, stop.
If there is an edge uv such that G − {u, v} is P3-free, then output S_A = {u, v}, stop.
Find a vertex set S which is an α-approximation solution to VCP3 on G.
if |L_S| = 1 then
    Output S_A ← S and stop.
end if
while |L_S| ≥ 2 and there exists r ∈ R_S with d_{G^c_{S,V\S}}(r) ≥ 3 do
    S ← S ∪ A_r
end while
Recursively use the operations in the proof of Lemma 2 to change S into S' satisfying the conditions of Lemma 2.
Apply Operation A on S' until no more Operation A can be applied.
if |L_S'| = 1 then
    Output S_A ← S' and stop.
end if
Find a minimum connected vertex cover C of G^c_{S',V\S'} containing L_S'.
Output S_A ← S' ∪ (⋃_{r∈C∩R_S'} A_r).
```

By Lemma 2, step 10 can be executed in time $O(n^2)$. By Remark 5, the running time for step 15 is $O(n^2+m)$, where m is the number of edges. In fact, the construction of H can be done in time $O(n^2)$, finding a spanning tree in H can be done in time $O(m)$, and adding those vertices $\{r_e\}_{e\in E(T)}$ needs time $O(n)$. As to step 1, step 11, and the while loop, it is easy to see that their running time is $O(n)$. The running time for step 2 is $O(mn)$. So, if we denote by M the running time for the α-approximation algorithm computing VCP_3, the time complexity of Algorithm 1 is $O(mn+M)$.

To analyze the performance ratio of Algorithm 1, we assume that S_A is output in line 16 (if S_A is output in line 1 or line 2, then it is an optimal solution, if S_A is output in line 5 or line 13, then its size will be smaller than the one output in line 16). The next lemma gives a lower bound for the size of an optimal $CVCP_3$.

Lemma 6. *Let S' be the vertex set found in line 11 of Algorithm 1. Then $opt_{CVCP_3}\geq 2|L_{S'}|-1$, where opt_{CVCP_3} is the size of a minimum $CVCP_3$.*

Proof. Let OPT be a minimum $CVCP_3$ of G, and let $OPT^c_{S',V\setminus S'}=\{s\in V(G^c_{S',V\setminus S'})\colon A_s\cap OPT\neq\emptyset\}$ (recall that A_s is the set of vertices in G corresponding to the vertex s of $G^c_{S',V\setminus S'}$). Consider Fig. 1 as an illustration, the dotted circle in Fig. 1(a) indicates the vertex set OPT in G, and the dotted circle in Fig. 1(b) indicates the vertex set $OPT^c_{S',V\setminus S'}$ in $G^c_{S',V\setminus S'}$. For simplicity of notation, denote by H the subgraph of $G^c_{S',V\setminus S'}$ induced by $OPT^c_{S',V\setminus S'}$. Notice that H can be viewed as a graph obtaining from $G[OPT]$ by a set of contractions, that is, collapsing all vertices in a same connected component of $G[S]$ or $G[V\setminus S']$ which has nonempty intersection with OPT onto one vertex of H. So, by the connectedness of $G[OPT]$, the graph H is also connected.

Hence if $OPT^c_{S',V\setminus S'}$ is a vertex cover of $G^c_{S',V\setminus S'}$, then it is a CVC of $G^c_{S',V\setminus S'}$, and thus $|OPT|\geq|OPT^c_{S',V\setminus S'}|\geq opt_{CVC(G^c_{S',V\setminus S'})}=2|L_{S'}|-1$, where $opt_{CVC(G^c_{S',V\setminus S'})}$ is the size of a minimum CVC of $G^c_{S',V\setminus S'}$ and the equality holds because of Lemma 4.

In the following, assume that $OPT^c_{S',V\setminus S'}$ is not a vertex cover of $G^c_{S',V\setminus S'}$. Then there exists an isolated edge uv in $G^c_{S',V\setminus S'}-OPT^c_{S',V\setminus S'}$. Notice that u and v are normal vertices (not contracted vertices) in G belonging to $L_{S'}$ and $R_{S'}$, respectively. This is because OPT is a VCP_3 of G and thus $G-OPT$ contains only isolated vertices and isolated edges. Denote by E' the set of isolated edges in $G^c_{S',V\setminus S'}-OPT^c_{S',V\setminus S'}$. In the following, when we mention an edge $uv\in E'$, it is always assumed that $u\in L_{S'}$ and $v\in R_{S'}$.

Notice that if $L_{S'}\cap OPT^c_{S',V\setminus S'}=\emptyset$, then $OPT\subseteq V\setminus S'$. Since $G[OPT]$ is connected and every connected component in $G[V\setminus S']$ is either a vertex or an edge, we have $|OPT|\leq 2$. In this case, the algorithm stops at the first line or the second line. Since we are now considering the set S' found in line 11, hence

$$L_{S'}\cap OPT^c_{S',V\setminus S'}\neq\emptyset. \tag{2}$$

Consider an edge $uv\in E'$. Let the neighbors of u in $G^c_{S',V\setminus S'}$ be v and $r_1^{(u)},\ldots,r_{t_u}^{(u)}$. Because $d_{G^c_{S',V\setminus S'}}(l)\geq 2$ holds for every vertex $l\in L_{S'}$, we have $t_u\geq 1$. Notice that $\{r_1^{(u)},\ldots,r_{t_u}^{(u)}\}\subseteq R_{S'}\cap OPT^c_{S',V\setminus S'}$ (otherwise there will be a P_3-path in $G[V\setminus OPT]$). Since every vertex in $R_{S'}$ has degree at most 2 in $G^c_{S',V\setminus S'}$, and because graph H (which is $G^c_{S',V\setminus S'}[OPT^c_{S',V\setminus S'}]$) is connected with $L_{S'}\cap V(H)\neq\emptyset$ (see (2)), so

$$\begin{aligned}&\text{every } r_i^{(u)} \text{ has degree exactly 2 in } G^c_{S',V\setminus S'} \text{ and}\\&\text{the other neighbor of } r_i^{(u)} \text{ belongs to } L_{S'}\cap OPT^c_{S',V\setminus S'}.\end{aligned} \tag{3}$$

Suppose that there exists an edge $uv\in E'$ such that $t_u=1$ and $|A_{r_1^{(u)}}|=1$. Let w be the unique vertex in $A_{r_1^{(u)}}$. Notice that w and v are the only two neighbors of u in G, and all neighbors of v belong to S'. Hence $S''=(S'\setminus\{u\})\cup\{w\}$ is still a VCP_3 of G. Since the other neighbor of vertex w is in S' and u is an isolated vertex in $G[S']$, hence $G[S'']$ has fewer connected components than $G[S']$. These imply that Operation A is still applicable on S', contradicting the construction of S' in line 11 of Algorithm 1. So,

$$\text{for any edge } uv\in E',\ \text{we have } \Big|\bigcup_{i=1}^{t_u}A_{r_i^{(u)}}\Big|\geq 2. \tag{4}$$

Call those vertices contained in $L_{S'}$ and $R_{S'}$ as L-vertices and R-vertices, respectively. Let $G':=G^c_{S',V\setminus S'}-\{u\colon uv\in E'\}-\{v\colon uv\in E'$ and u is the unique neighbor of $v\}$ be the graph obtained from $G^c_{S',V\setminus S'}$ by removing all L-vertices incident to the edges of E' and those R-vertices incident to the edges of E' which have degree one in G. Then G' is connected and $OPT^c_{S',V\setminus S'}$ is a CVC of G'. Notice that G' satisfies the condition of Lemma 4. It should be pointed out that if we remove all end vertices incident to the edges of E', then G' is connected (since all isolated vertices and isolated edges are linked to the connected subgraph $G^c_{S',V\setminus S'}[OPT^c_{S',V\setminus S'}]$), but the degree of some vertices in the L-part might become smaller than two. This is why we only remove those R-vertices of degree one in G. By observation (3), every $r_i^{(u)}$ is a leaf vertex of G' (and also a leaf vertex of H), hence $OPT^c_{S',V\setminus S'}-\bigcup_{uv\in E'}\left(\bigcup_{i=1}^{t_u}\{r_i^{(u)}\}\right)$ is still a CVC of G'. So,

$$\left|OPT^c_{S',V\setminus S'}-\bigcup_{uv\in E'}\left(\bigcup_{i=1}^{t_u}\{r_i^{(u)}\}\right)\right|\geq opt_{CVC(G')}=2(|L_{S'}-\{u\colon uv\in E'\}|)-1.$$

By noticing that the sets $\{A_{r_i}^{(u)}\}_{u:uv\in E'}$ are mutually disjoint, we have

$$|OPT| \geq \left| OPT^c_{S',V\setminus S'} - \bigcup_{uv\in E'} \left(\bigcup_{i=1}^{t_u} \{r_i^{(u)}\} \right) \right| + \sum_{uv\in E'} \left| \bigcup_{i=1}^{t_u} A_{r_i^{(u)}} \right|.$$

Then by observation (4),

$$|OPT| \geq 2(|L_{S'}| - |E'|) - 1 + 2|E'| = 2|L_{S'}| - 1.$$

The lemma is proved. □

Next, we analyze the performance ratio of Algorithm 1.

Theorem 7. *Algorithm 1 has performance ratio at most* $2\alpha + 1/2$.

Proof. Let OPT be a minimum $CVCP_3$ of G, and let $opt = |OPT|$. Denote the vertex set found in line 3 as S_1, the vertex set obtained at the end of the while loop as S_2. Since S_1 is an α-approximation to $MinVCP_3$ and the size of a minimum VCP_3 is no larger than the size of a minimum $CVCP_3$, we have

$$|S_1| \leq \alpha \cdot opt.$$

Suppose the while loop is executed t times. Because every $|A_r| \leq 2$ and thus each iteration adds at most two vertices, so

$$|S_2| \leq |S_1| + 2t.$$

Since each iteration merges at least three connected components into one larger connected component, we have

$$|L_{S_1}| \geq |L_{S_2}| + 2t.$$

After line 11 of Algorithm 1, the set S' satisfies

$$|S'| \leq |S_2| \text{ and } |L_{S'}| \leq |L_{S_2}|.$$

By Lemma 4 and Remark 5,

$$|S_A| \leq |S'| + 2(|L_{S'}| - 1),$$

where coefficient 2 comes from the fact that each A_r has cardinality at most 2. Combining the above inequalities together, and using the fact $|L_{S_1}| \leq |S_1|$, we have

$$|S_A| \leq |S_1| + |L_{S_1}| - |L_{S_2}| + 2|L_{S'}| - 2 \leq 2|S_1| + |L_{S'}| - 2.$$

By Lemma 6,

$$opt \geq 2|L_{S'}| - 1.$$

Hence

$$|S_A| \leq \left(2\alpha + \frac{1}{2}\right) opt - \frac{3}{2}.$$

The theorem is proved. □

4. Conclusion and discussion

In this paper, we give a polynomial-time $(2\alpha + 1/2)$-approximation algorithm for $MinCVCP_3$, where α is the performance ratio of a polynomial-time algorithm for $MinVCP_3$. From [11,20], it is known that $MinCVCP_3$ can be approximated within factor 3. For those classes of graphs on which $MinVCP_3$ has approximation factor $\alpha < 5/4$, the algorithm in this paper is a kind of improvement. It deserves to be further studied whether the coefficient 2 before α can be reduced to some constant strictly smaller than 2. Designing approximation algorithms for $MinVCP_3$ with small performance ratio α in some graph classes is another research topic of our future study.

Another question is whether there exists a class of graphs for which $MinVCP_3$ is polynomial-time solvable but $MinCVCP_3$ is NP-hard. We have explored many typical classes of graphs but have not found one of them having such a disparate. Finding such a class of graphs might be an interesting research topic. Furthermore, bounds (especially lower bounds) of approximation ratio for $MinVCP_k$ and $MinCVCP_k$ in various graph classes are not sufficiently studied yet.

The above question also motivates us to ask the question of *price of connectivity for approximation*. In [6], Cardinal and Levy introduced the concept of *price of connectivity* for the vertex cover problem, which is the worst-case ratio between the sizes of a minimum connected vertex cover and a minimum vertex cover. Here we are interested in a similar question from an approximation point of view: if the approximation ratio of a minimization problem is α, and its connected version has approximation ratio β, what can we say about the ratio β/α, for the considered graph class?

Here, approximation ratio is the infimum of r such that the problem is r-approximable in polynomial time. We abbreviate such a ratio as PoC-APPROX. For example, both the minimum vertex cover problem (MinVC) and the minimum connected vertex cover problem (MinCVC) have 2-approximations which are widely believed to be tight, so PoC-APPROX(VC) might be 1. In [9], Escoffier et al. proved that MinCVC is APX-hard on bipartite graphs and presented a 5/3-approximation algorithm for those classes of graphs on which MinVC is polynomial-time solvable. Since MinVC is polynomial-time solvable on bipartite graphs, PoC-APPROX(VC) for bipartite graphs is strictly larger than 1 and upper bounded by 5/3. By the statement in the second paragraph of Section 1.2, PoF-APPROX(VCP_3) has a trivial upper bound 3. Our result implies that PoF-APPROX(VCP_3) ≤ 2.5, which improves the above trivial bound 3. New techniques are needed for further improvement.

CRediT authorship contribution statement

Pengcheng Liu: Formal analysis, Investigation, Writing - original draft. **Zhao Zhang:** Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Writing - review & editing. **Xianyue Li:** Methodology, Validation, Writing - review & editing. **Weili Wu:** Conceptualization, Project administration, Validation.

Acknowledgments

This research is supported by NSFC (11771013, 11531011, 61751303) and ZJNSFC, China (LD19A010001), the Fundamental Research Funds for the Central Universities, China (No. lzujbky-2017-163), and NSF, USA (#1747818).

References

[1] R. Boliac, K. Cameron, V. Lozin, On computing the dissociation number and the induced matching number of bipartite graphs, Ars Combin. 72 (2004) 241–253.
[2] B. Brešar, F. Kardoš, J. Katrenič, G. Semanišin, Minimum k-path vertex cover, Discrete Appl. Math. 159 (2011) 1189–1195.
[3] B. Brešar, T. Kos, R. Krivoš-Belluš, G. Semanišin, Hitting subgraphs in P_4-tidy graphs, Appl. Math. Comput. 352 (2019) 211–219.
[4] B. Brešar, R. Krivoš-Belluš, G. Semanišin, P. Šparl, On the weighted k-path vertex cover problem, Discrete Appl. Math. 177 (2014) 14–18.
[5] K. Cameron, P. Hell, Independent packings in structured graphs, Math. Program. 105 (2–3) (2006) 201–213.
[6] J. Cardinal, E. Levy, Connected vertex covers in dense graphs, Theoret. Comput. Sci. 411 (26–28) (2010) 2581–2590.
[7] M.S. Chang, L.H. Chen, L.J. Hung, P. Rossmanith, P.C. Su, Fixed-parameter algorithms for vertex cover P_3, Discrete Optim. 19 (2016) 12–22.
[8] L. Chen, X. Huang, Z. Zhang, A simpler PTAS for connected k-path vertex cover in homogeneous wireless sensor network, J. Combin. Optim. 36 (1) (2018) 35–43.
[9] B. Escoffier, L. Gourvès, J. Monnot, Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs, J. Discrete Algorithms 8 (2010) 36–49, A preliminary version appears in International Workshop on Graph-Theoretic Concepts in Computer Science 2007, LNCS 4769 (2007) 202–213.
[10] T. Fujito, A unified approximation algorithm for node-deletion problems, Discrete Appl. Math. 86 (1998) 213–231.
[11] T. Fujito, On approximability of connected path vertex cover, in: Workshop on Approximation and Online Algorithms, 2017, pp. 17–25.
[12] M. Jakovac, A. Taranenko, On the k-path vertex cover of some graph products, Discrete Math. 313 (2013) 94–100.
[13] F. Kardoš, J. Katrenič, I. Schiermeyer, On computing the minimum 3-path vertex cover and dissociation number of graphs, Theoret. Comput. Sci. 412 (2011) 7009–7017.
[14] J. Katrenič, A faster FPT algorithm for 3-path vertex cover, Inform. Process. Lett. 116 (4) (2016) 273–278.
[15] S. Khot, D. Minzer, M. Safra, Pseudorandom sets in Grassmann graph have near-perfect expansion, in: FOCS2018, pp. 592–601.
[16] M.S. Krishnamoorthy, N. Deo, Node-deletion NP-complete problems, SIAM J. Comput. 8 (4) (1979) 619–625.
[17] E. Lee, Partitioning a graph into small pieces with applications to path transversal, Math. Program. Ser. A (2019) 1546–1558, http://dx.doi.org/10.1007/s10107-018-1255-7, A preliminary version appears in SODA 2017.
[18] J.M. Lewis, M. Yannakakis, The node-deletion problem for hereditary properties is NP-complete, J. Comput. System Sci. 20 (1980) 219–230.
[19] Y. Li, J. Tu, A 2-approximation algorithm for the vertex cover P_4 problem in cubic graphs, Int. J. Comput. Math. 91 (10) (2014) 2103–2108.
[20] X. Li, Z. Zhang, X. Huang, Approximation algorithms for minimum (weight) connected k-path vertex cover, Discrete Appl. Math. 205 (2016) 101–108.
[21] X. Liu, H. Lu, W. Wang, W. Wu, PTAS for the minimum k-path connected vertex cover problem in unit disk graphs, J. Global Optim. 56 (2013) 449–458.
[22] M. Novotný, Design and analysis of a generalized canvas protocol, in: Proceedings of WISTP 2010, in: LNCS, vol. 6033, 2010, pp. 106–121.
[23] Y. Orlovich, A. Dolgui, G. Finke, V. Gordon, F. Werner, The complexity of dissociation set problems in graphs, Discrete Appl. Math. 159 (2011) 1352–1366.
[24] Y. Ran, Z. Zhang, X. Huang, X. Li, D. Du, Approximation algorithms for minimum weight connected 3-path vertex cover, Appl. Math. Comput. (2019) 723–733.
[25] J. Tu, A fixed-parameter algorithm for the vertex cover P_3 problem, Inform. Process. Lett. 115 (2015) 96–99.
[26] J. Tu, F. Yang, The vertex cover P_3 problem in cubic graphs, Inform. Process. Lett. 113 (2013) 481–485.
[27] J. Tu, W. Zhou, A factor 2 approximation algorithm for the vertex cover P_3 problem, Inform. Process. Lett. 111 (2011) 683–686.
[28] J. Tu, W. Zhou, A primal–dual approximation algorithm for the vertex cover P_3 problem, Theoret. Comput. Sci. 412 (2011) 7044–7048.
[29] H. Vogt, Exploring message authentication in sensor networks, in: ESAS2004, in: LNCS, vol. 3313, 2015, pp. 19–30.
[30] Y. Xiao, X. Du, A survey on sensor network security, in: Y. Li, et al. (Eds.), Wireless Sensor Networks and Applications, in: Signals and Communication Technology Series, Springer, Heidelberg, 2008, pp. 403–421.
[31] M. Xiao, S. Kou, Exact algorithms for the maximum dissociation set and minimum 3-path vertex cover problems, Theoret. Comput. Sci. 657 (2017) 86–97.
[32] J. You, J.X. Wang, Y.X. Cao, Approximate association via dissociation, Discrete Appl. Math. 219 (2017) 202–209.
[33] A. Zhang, Y. Chen, Z.-Z. Chen, G. Lin, Improved approximation algorithms for path vertex covers in regular graphs, arXiv:1811.01162v1.
[34] Z. Zhang, X. Li, Y. Shi, H. Nie, Y. Zhu, PTAS for minimum k-path vertex cover in ball graph, Inform. Process. Lett. 119 (2017) 9–13.
[35] Y. Zhang, Y. Shi, Z. Zhang, Approximation algorithm for the minimum weight connected k-subgraph cover problem, Theoret. Comput. Sci. 535 (2014) 54–58.