The connected vertex cover problem in k-regular graphs

Yuchao Li[1] · Wei Wang[1] · Zishen Yang[1]

Abstract

Given a connected graph $G = (V, E)$, the Connected Vertex Cover (CVC) problem is to find a vertex set $S \subset V$ with minimum cardinality such that every edge is incident to a vertex in S, and moreover, the induced graph $G[S]$ is connected. In this paper, we investigate the CVC problem in k-regular graphs for any fixed k ($k \geq 4$). First, we prove that the CVC problem is NP-hard for k-regular graphs, and then we give a lower bound for the minimum size of a CVC, based on which, we propose a $\frac{2k}{k+2} + O(\frac{1}{n})$-approximation algorithm for the CVC problem.

Keywords Connected vertex cover problem · Approximation algorithm · k-Regular graph

Mathematics Subject Classification 05C85 · 05C70

1 Introduction

All the graphs considered in this paper are simple graphs without loops and multi-edges. Given a graph G, $V(G)$ and $E(G)$ denote the vertex set and edge set of G, respectively. For a subset $X \subseteq V(G)$, $|X|$ denotes the size of X which is the number of vertices of X. $G[X]$ is the subgraph induced by X. For $v \in V(G)$, *the degree $d(v)$* is equal to the number of neighbors of v. $\Delta(G)$ denotes the maximum degree of G and $\Delta(G) = \max_{v \in V(G)} d(v)$. If all the vertices of G have the same degree k, then G is called a k-regular graph. A connected graph is a graph in which any two vertices

✉ Wei Wang
    wang_weiw@163.com

    Yuchao Li
    liyu_chao@126.com

    Zishen Yang
    yangzishen@sina.com

[1] School of Mathematics and Statistics, Xi'an Jiaotong University, No. 28 Xianning West Road,
    Xi'an 710049, Shaanxi, China

are connected to each other by a path. A graph in which each pair of distinct vertices is joined by an edge is called a complete graph. A complete graph with n vertices is denoted by K_n. A clique of a simple graph G is a subset S of V such that $G[S]$ is complete. A bipartite graph is one whose vertex set can be partitioned into two subsets X and Y, so that each edge has one end in X and one end in Y; such a partition (X, Y) is called a bipartition of the graph. A complete bipartite graph is a bipartite graph with bipartition (X, Y) in which each vertex of X is joined to each vertex of Y; if $|X| = m$ and $|Y| = n$, such a graph is denoted by $K_{m,n}$. For $X, Y \subseteq V(G)$, the set of edges between X and Y is denoted by $E(X, Y)$.

The vertex cover (VC) *problem* is a classic problem in combinatorial optimization and operations research. Given a graph G, the VC problem aims to find a minimum vertex subset $C \subset V$ (vertex cover set) such that every edge in the graph is incident to a vertex in C. *The connected vertex cover* (CVC) *problem* is a variation of the vertex cover problem, which is to find a vertex cover set F with minimum cardinality such that $G[F]$ is connected.

The CVC problem was first introduced by Garey and Johnson (1977), which finds many applications in real life. For example, in the field of wireless network design (Moser 2005), the vertices and the edges represent the network nodes and transmission links, respectively. Some relay stations will be placed on some network nodes such that they form a connected subnetwork and every transmission link is incident to a relay station. The target is to minimize the number of relay stations. This is exactly the connected vertex cover problem.

The CVC problem is NP-hard (Garey and Johnson 1979) in general graphs. Savage (1982) and Fujito and Doi 2004 have proposed two 2-approximation algorithms for the connected vertex cover problem, respectively. But this problem is NP-hard to be approximated within ratio $10\sqrt{5} - 21$ (Fernau and Manlove 2006). Recently the fixed-parameter tractability of the connected vertex cover problem with respect to the vertex cover size or to the treewidth of the input graph has been widely studied; see e.g., Fernau and Manlove (2006), Guo et al. (2005), Mölle et al. (2008, 2006) and Moser (2005).

Also, a great deal of efforts have been devoted to this problem on various classes of graphs. The researchers proved that it is NP-complete in planar graphs of maximum degree 4 (see Garey and Johnson 1977), in planar bipartite graphs of maximum degree 4 (see Fernau and Manlove 2006), in planar biconnected graphs of maximum degree 4 (see Priyadarsini and Hemalatha 2008) and in 3-connected graphs (see Wanatabe et al. 1991). In Esc et al. (2010), the authors showed that the connected vertex cover problem is polynomial-time solvable in chordal graphs and proved that the problem is APX-complete in bipartite graphs of maximum degree 4, even if each vertex of one block of the bipartition has a degree at most 3. On the other hand, if each vertex of one block of the bipartition has a degree at most 2 (and the vertices of the other part have an arbitrary degree), then the problem is polynomial time solvable. They also showed that the connected vertex cover problem is $\frac{5}{3}$-approximable in any class of graphs where the vertex cover problem is polynomial time solvable (in particular in bipartite graphs, or more generally in perfect graphs). Then, they presented a polynomial approximation scheme for the connected vertex cover problem in planar graphs. Cardinal and Levy (2010) gave new approximation results for the connected vertex cover problem

in dense graphs, in which either the minimum or the average degree is linear. In particular, they proved tight parameterized upper bounds on the approximation returned by Savages algorithm (1982), and extended a vertex cover algorithm from Karpinski and Zelikovsky (1997) to the connected case. The new algorithm approximates the minimum connected vertex cover problem within a factor strictly less than 2 on all dense graphs. Also, Zhang et al. (2009) gave the first polynomial time approximation scheme for the connected vertex cover problem in unit disk graphs.

Ueno et al. (1988) proved that the connected vertex cover problem can be solved in polynomial time for graphs with no vertex degree exceeding 3. Thus, this problem can be polynomial-time solved for 3-regular graph. But, Li et al. (2017) proved that the connected vertex cover problem is NP-hard on 4-regular graphs. They also presented a $\frac{4}{3} + O(\frac{1}{n})$-approximation algorithm and compare it to the 2-approximation algorithm in Fujito and Doi (2004).

In this paper, we shall investigate the CVC problem in k-regular graphs. We proved that the CVC problem is NP-hard in k-regular graphs for any fixed k ($k \geq 4$) and gave a lower bound for the minimum size of a CVC. Then we proposed an approximation algorithm with approximation ratio $\frac{2k}{k+2} + O(\frac{1}{n})$ for the CVC problem in k-regular graphs.

In Sect. 2, we show that CVC is NP-hard in k-regular graphs for any fixed $k(k \geq 4)$. In Sect. 3, we present a lower bound for CVC problem. In Sect. 4, we propose a $\frac{2k}{k+2} + O(\frac{1}{n})$-approximation algorithm. Conclusions and future works are given in Sect. 5.

2 The connected vertex cover problem is NP-hard in k-regular graphs for any fixed k ($k \geq 4$)

In this section, we show the NP-hardness of the CVC problem on k-regular graphs, for any fixed $k \geq 4$.

Theorem 1 *The CVC problem is NP-hard in k-regular graphs for any fixed $k(k \geq 4)$.*

Proof We reduce the CVC problem for 4-regular graphs, which is proved to be NP-hard (Li et al. 2017), to the CVC problem in k-regular graphs for any fixed $k(k \geq 5)$.

Given a 4-regular graph H, we construct a new graph G in the following way. For every vertex $u \in V(H)$ (adjacent to α, β, γ and δ), split it into a four cycle $u_1 u_2 u_3 u_4$ with 4 edges $u_1\alpha, u_2\beta, u_3\gamma$ and $u_4\delta$. Then add a clique K_{k-3} with vertices labeled $u_5, u_6, \ldots, u_{k+1}$. At last, connect vertices u_i ($i = 1, 2, 3, 4$) and vertices u_j ($j = 5, 6, \ldots, k + 1$) by edges $u_i u_j$. Hence, all the vertices in graph G has degree k, see Fig. 1.

Let $S \subseteq V(H)$ be a connected vertex cover set in H. Let $S' = \{u_1, u_2, \ldots, u_k | u \in S\} \cup \{u_2, u_4, u_5, \ldots, u_{k+1} | u \in V \backslash S\}$. S' is a connected vertex cover set in G and $|S'| = |S| + (k - 1)|V(H)|$.

Conversely, let S' be a connected vertex cover set in G. It is obvious that at most two vertices of $\{u_1, u_2, \ldots, u_{k+1} | u \in V(H)\}$ are not in S'. Let $S'' = \{u_1, u_2, \ldots, u_k | |S' \cap \{u_1, u_2, \ldots, u_{k+1}\}| \geq k, u \in V(H)\} \cup \{u_2, u_4, u_5, \ldots, u_{k+1} | |S' \cap \{u_1, u_2, \ldots, u_{k+1}\}| = k - 1, u \in V(H)\}$. Thus, we construct another connected vertex cover set $S'' \subseteq$
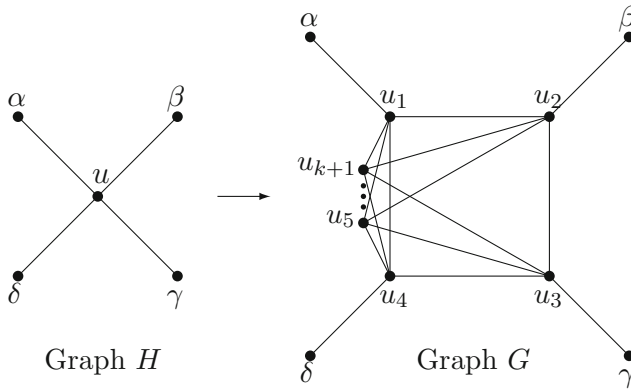
Fig. 1 The transformation in the reduction from a 4-regular graph to a k-regular graph

$V(G)$. Obviously S'' is also a connected vertex cover set in G and $|S''| \leq |S'|$. Let $S = \{u \mid |S'' \cap \{u_1, u_2, \ldots, u_{k+1}\}| = k, u \in V(H)\}$. Clearly, S is a connected vertex cover set in H and $|S| = |S''| - (k-1)|V(H)|$.

The conclusion follows since G can be constructed in polynomial time from H. □

3 A lower bound

In this section, we give a sharp lower bound for the minimum size of a CVC.

Theorem 2 *Let G be a k-regular graph of order n, if S'_{opt} is a minimum connected vertex cover set of G, then $|S'_{opt}| \geq \frac{kn-2}{2k-2}$ and this lower bound is sharp.*

Proof Let S' be a connected vertex cover set of G. Since $G[S']$ is connected, $E(G[S']) \geq |S'| - 1$. So there are at most $k|S'| - 2(|S'| - 1)$ edges between S' and $V \setminus S'$. On the other hand, $E(G[V \setminus S']) = 0$ otherwise there must exist uncovered edge. So $E(S', V \setminus S') = k|V \setminus S'|$. Thus we have

$$E(S', V \setminus S') = k|V \setminus S'| \leq k|S'| - 2(|S'| - 1).$$

Since $|V \setminus S'| = n - |S'|$, we obtain $|S'| \geq \frac{kn-2}{2k-2}$. □

The example in Fig. 2 shows the lower bound is sharp. In this graph, u is connected to $\{u_1, u_{(n-1)k+2}, u_{(n-1)k+3}, \ldots, u_{nk}\}$ and $u_1 u_2 \ldots u_{(n-1)k} u_{(n-1)k+1}$ is a path. In each rectangle, the vertices and the edges between u_i and v_j construct a complete bipartite graph. Clearly the graph in Fig. 2 is a k-regular graph with $nk+1+(n-1)(k-2)+k-1 = 2(k-1)n+2$ vertices and $\{u, u_1, u_2, \ldots, u_{nk}\}$ is a connected vertex cover set in this graph. Since any vertex cover set in this graph has at least $\frac{k \times [2(k-1)n+2]-2}{2k-2} = nk+1$ vertices, $\{u, u_1, u_2, \ldots, u_{nk}\}$ is the minimum connected vertex cover set. This implies that the lower bound is sharp.
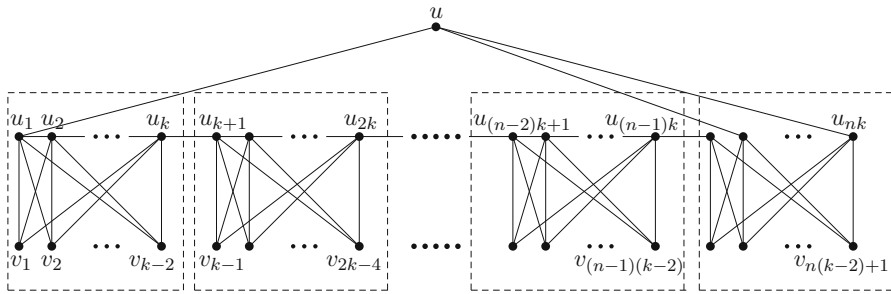
Fig. 2 An example for Theorem 2

4 A $\frac{2k}{k+2} + O\left(\frac{1}{n}\right)$-approximation algorithm

In this section, we shall present a polynomial time approximation algorithm for the CVC problem on k-regular graphs, for any fixed $k \geq 4$.

For the convenience of the reader, first we give some definitions.

Recall that *a polynomial time ρ-approximation algorithm $\mathcal{A}$* for a minimization problem is an algorithm whose running time is bounded by a polynomial in the size of the input and always outputs a feasible solution with objective value guaranteed to be within ρ times the optimal one.

A subset S of $V(G)$ is called an independent set of G if no two vertices of S are adjacent in G. A set $S \subseteq V$ is an independent set of G if and only if $V \setminus S$ is a vertex cover set of G. So if we find an independent set F such that $G[V \setminus F]$ is connected, then $V \setminus F$ is a connected vertex cover set in G.

A vertex v of G is a cut vertex if E can be partitioned into two nonempty subsets E_1 and E_2 such that $G[E_1]$ and $G[E_2]$ have just the vertex v in common. A connected graph that has no cut vertices is called a block. A block of a graph is a subgraph that is a block and is maximal with respect to this property. Every graph is the union of its blocks.

Now we briefly describe our algorithm. Given a connected graph G. First, we find all blocks $\{B_1, B_2, \ldots, B_s\}$ in G. For each block $B_i (i = 1, 2, \ldots, s)$, if there is a k-degree vertex v in it, we delete v from graph G. After this round, we obtain a new graph, still denoted by G, then repeat above process, until either there is no blocks in G or no k-degree vertex left in each block. Algorithm 1 is a formal description of our algorithm.

Theorem 3 *The vertex set F computed by Algorithm 1 is a connected vertex cover set in a k-regular graph G.*

Proof For each block in current graph, we choose a k-degree vertex if there exists one. This ensures that the chosen vertices are not cut vertices. So, deleting them from current graph can still make the remaining graph connected. Thus we update the graph by deleting the chosen vertices from the current graph. Notice that each two chosen vertices are not adjacent. Then I is a independent set. This implies that $F = V(G) \setminus I$ is a vertex cover set in G. Thus, the vertex set F computed by Algorithm 1 is a connected vertex cover set since $G[F] = G[V(G) \setminus I]$ is connected. $\qquad\square$

Algorithm 1 A $\frac{2k}{k+2} + O(\frac{1}{n})$-approximation algorithm

---

Require: A k-regular graph $G = (V, E)$
Ensure: A connected vertex cover set F
1: $I = \emptyset$, $G' = G$
2: **while** there exists block in G' **do**
3:     Compute all the blocks $B = \{B_1, B_2, \ldots, B_k\}$ in G'
4:     **for** $i = 1, 2, \ldots, k$ **do**
5:         **if** there exists k-degree vertex in B_i **then**
6:             Find a k-degree vertex v in B_i
7:             $G' = G' \backslash \{v\}$, $I = I \cup \{v\}$
8:         **end if**
9:     **end for**
10: **end while**
11: Output $F = V(G) \backslash I$

The following lemma is a key to the analysis of the performance of Algorithm 1.

Lemma 4 *Given a connected simple graph G with maximum degree $\Delta \leq k$, and all k-degree vertices are cut vertices. Then*

$$2|E(G)| = \sum_{v \in V(G)} d(v) < \frac{k^2 - k + 2}{k}|V(G)|.$$

To prove this lemma, we first introduce two definitions to illustrate the connectivity and the structure of the given graph in Lemma 4.

Definition 1 (Diestel 2000) The block graph of a graph G is defined as follows:

- All vertices in block graph represent blocks and cut vertices in the original graph G.
- There is an edge in block graph if and only if the corresponding cut vertex is in the corresponding block.

Proposition 5 (Diestel 2000) *The block graph of a connected graph is actually a tree.*

It is usually called the block tree instead of the block graph if the base graph G is connected, and notation $T_B(G)$ is used to represent the structure tree in this case. Sometimes what we concern about is not all blocks or the structure of the blocks, but the structure of block groups of the original graph. Thus, following definition of i-block tree comes out.

Definition 2 The i-block tree ($i \in \mathbb{N}_+$) $T_i(G)$ of a connected graph G is a tree obtained from the original graph G by following steps:

- Find out all blocks and all cut vertices in G to build a block tree $T_B(G)$.
- Contract all edges incident to a vertex, which is corresponding to a cut vertex whose degree is not i, in tree $T_B(G)$.
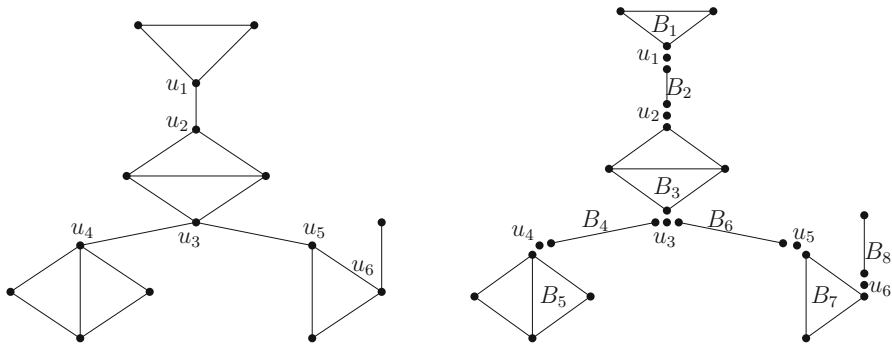- Simplify the graph by loop and multiple edge deletion.

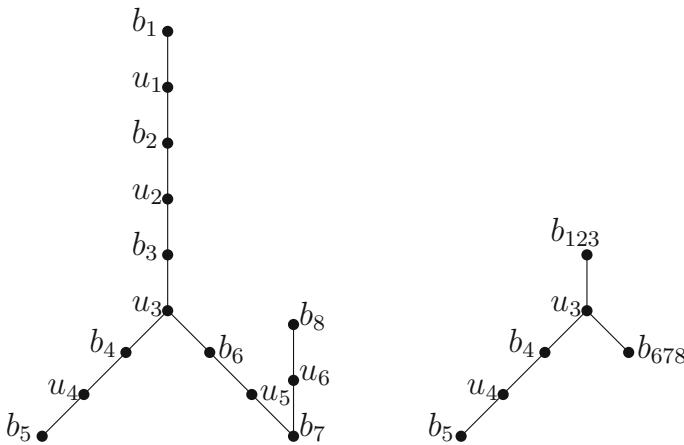**Fig. 3** Graph G (left) and all blocks and cut vertices in G (right)

Fig. 4 Block tree $T_B(G)$ of G (left) and 4-block tree $T_4(G)$ of G (right)

Actually, i-block tree is a real tree since it is created by edge contraction and simplification from a tree. We denote this kind of trees by $T_i(G)$ for original connected graph G. Especially, $T_k(G)$ gives the structure of k-degree cut vertices and block groups.

Figures 3 and 4 show the creation of block tree and 4-block tree of connected graph G. The left graph in Fig. 3 is a connected graph G with $\Delta = 4$ and the right graph demonstrates all its cut vertices and blocks. The left graph in Fig. 4 shows the block tree $T_B(G)$ of graph G. In $T_B(G)$, u_3, u_4 represent 4-degree cut vertices in graph G and u_1, u_2, u_5, u_6 are 3-degree cut vertices in G. $b_i (i = 1, 2 \ldots, 8)$ represent the eight blocks in graph G. The right graph in Fig. 4 is the 4-block tree $T_4(G)$ of graph G.

Next, we will prove Lemma 4.

Proof Let $\lambda(G)$ and $\mu(G)$ denote the number of k-degree vertices and $(k-1)$-degree vertices in graph G, respectively. We will prove the lemma by induction on $\lambda(G)$.
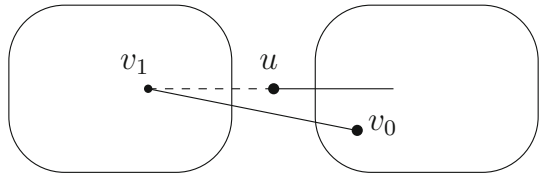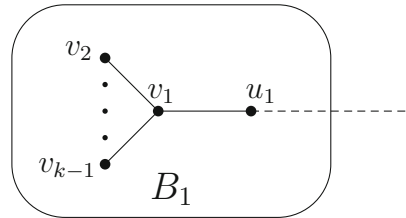
Fig. 5 The transformation from G to G'

Fig. 6 The block or may be a group of blocks B_1 in graph G

The conclusion is obviously when $\lambda(G) = 0$ since the maximum degree is $k - 1$ in the graph. We assume $2|E(G)| < \frac{k^2-k+2}{k}|V(G)|$ for all $\lambda(G) \leq l$. Then we will consider the situation $\lambda(G) = l + 1$.

Case 1 $\exists v_0 \in V(G), d(v_0) \leq k - 2$.

Let u be a k-degree cut vertex and v_1 is one of its neighbors, which belongs to different connected component in $G - u$ with v_0. Transform the graph G to G' by deleting the edge (u, v_1) and adding an edge (v_0, v_1) as Fig. 5 shows. Namely, $G' - (v_1, v_0) = G - (u_1, v_1)$. Therefore, $\lambda(G') = \lambda(G) - 1 = l$ and the induction works in this case.

Case 2 $\forall v \in V(G), d(v) \geq k - 1$.

In this case, the vertex in graph G is either $(k - 1)$-degree or k-degree, and all k-degree vertices are cut vertices. The statement of induction can be modified slightly as the number of k-degree vertices is less than $\frac{2}{k-2}$ times the number of $(k-1)$-degree vertices.

Find a leaf b_1 in the k-block tree $T_k(G)$. Denote its corresponding block or may be a group of blocks and adjacent cut vertex by B_1 and u_1, respectively. Note that the degree of the cut vertex $d_G(u_1) = k$ in original graph G, according to the definition of k-block tree. Obviously, there exists a vertex in B_1 adjacent to the cut vertex u_1 in graph G. Denote the vertex by v_1. Thus, resulting from the property of graph G, $d_G(v_1) = k - 1$ and all the neighbors of v_1 besides u_1 are in B_1 as it shows in Fig. 6. Namely, the number of $(k - 1)$-degree vertices in $V(B_1)$ is $|V(B_1 - u_1)| \geq k - 1$.

On the other hand, let $G' := G - (V(B_1)\backslash\{u_1\})$ be the graph obtained by deleting the block group B_1 but not the cut vertex u_1 in graph G. Hence, $\mu(G')+k-2 \leq \mu(G)$ and $\lambda(G') = \lambda(G) - 1 = l$ so that $\lambda(G') < \frac{2}{k-2}\mu(G')$ by induction. Therefore,

$$\lambda(G) = \lambda(G') + 1$$
$$< \frac{2}{k-2}\mu(G') + 1$$
$$\leq \frac{2}{k-2}(\mu(G) - k + 2) + 1$$

$$\lim_{n\to\infty} \frac{k \times (2n) + (k-1) \times (k-2)n + 2}{kn+2} = \frac{k^2 n - kn + 2n + 2}{kn+2} = \frac{k^2 - k + 2}{k}.$$
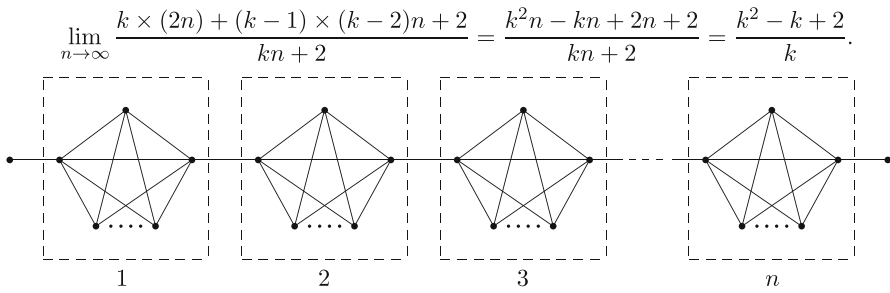

Fig. 7 Sharp example of Lemma 4

$$= \frac{2}{k-2}\mu(G) - 1$$
$$< \frac{2}{k-2}\mu(G).$$

That completes the proof.   □

The group of graphs in Fig. 7 gives a sharp example of Lemma 4. In these graphs, The vertices in each rectangle induce a complete graph K_k. Then, the asymptotically average degree is

$$\lim_{n\to\infty} \frac{k \times (2n) + (k-1) \times (k-2)n + 2}{kn+2} = \frac{k^2 n - kn + 2n + 2}{kn+2} = \frac{k^2 - k + 2}{k}.$$

Theorem 6 *Algorithm 1 is a $\frac{2k}{k+2} + O(\frac{1}{n})$-approximation algorithm which runs in $O(n^3)$ time.*

Proof When Algorithm 1 stops, either there is no blocks or no k-degree vertices in each block in the current graph.

In the first case, there is only one vertex v in the current graph. The vertex v has k neighbors $u_1, u_2, \ldots, u_k$ since the input is a k-regular graph. So there are at least $k + 1$ vertices in the input graph. By Theorem 2, there are at least three vertices ($\frac{k(k+1)-2}{2k-2} = \frac{k+2}{2} \geq 3$) in the connected vertex cover set. But in this case, only one vertex is in the connected vertex cover set. This is a contradiction. So this situation does not happen.

It follows that there are no k-degree vertices in each block in the current graph $G[F]$, then $G[F]$ is a graph with maximum degree k and all the k-degree vertices are cut vertices. We consider the number of edges in $G[F]$. By Lemma 4, $2|E(G[F])| < \frac{k^2-k+2}{k}|V(G[F])|$.

Now, let us compute the total number of edges in the input graph G. We have

$$E(G[F]) + k(n - |F|) = \frac{k}{2}n$$

$$\frac{k^2 - k + 2}{2k}|F| + k(n - |F|) > \frac{k}{2}n$$

$$|F| < \frac{k^2}{k^2 + k - 2}n.$$

So, the size of output set F of Algorithm 3 is less than $\frac{k^2}{k^2+k-2}n$. By Theorem 2, we have

$$\frac{|F|}{|S'_{opt}|} < \frac{\frac{k^2}{k^2+k-2}n}{\frac{kn-2}{2k-2}}$$

$$= \frac{2k}{k+2} + O\left(\frac{1}{n}\right)$$

Now we analyze the running time of Algorithm 1. Computing all the blocks in current graph requires $O(n)$ time. Finding a k-degree vertex in one block Step 6 can be done in $O(n)$ time. Since there are $O(n)$ blocks, Step 4 to Step 9 runs in $O(n^2)$ time. Thus, Algorithm 1 runs in $O(n^3)$ time.   □

5 Conclusions and future work

In this paper, we have investigated the connected vertex cover (CVC) problem in k-regular graphs. We proofed the CVC problem is NP-hard in k-regular graphs for any fixed $k(k \geq 4)$ and gave a sharp lower bound for it. Then we presented a $\frac{2k}{k+2} + O(\frac{1}{n})$-approximation algorithm for CVC problem in k-regular graphs which is better than the 2-approximation algorithm in Fujito and Doi (2004). As a future work, we will study the CVC problem in some other special classes of graphs and design approximation algorithms with better approximation ratio or efficient algorithms for it. Also there is a variation of the connected vertex cover problem: the connected vertex cover P_k problem. The connected vertex cover P_k ($CVCP_k$) problem is to find a minimum vertex set F which is additional required to induce a connected subgraph in a given connected graph G such that every path of order k in G contains at least one vertex from F. We will explore this problem in general graphs and special graphs in the future.

References

Cardinal J, Levy E (2010) Connected vertex covers in dense graphs. Theor Comput Sci 411:2581–2590
Diestel R (2000) Graph theory. Springer, New York, pp 43–44

Esc B, Gou L, Mon J (2010) Complexity and approximation results for the connected vertex cover problem in graphs and hypergraphs. J Discrete Algorithm 8:36–49

Fernau H, Manlove D (2006) Vertex and edge covers with clustering properties: complexity and algorithms. Algorithms Complex Durham 7:69–84

Fujito T, Doi T (2004) A 2-approximation NC algorithm for connected vertex cover and tree cover. Inf Process Lett 90:59–63

Garey MR, Johnson DS (1977) The rectilinear Steiner tree problem is NP-complete. SIAM J Appl Math 32:826–834

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. W. H. Freeman, London

Guo J, Niedermeier R, Wernicke S (2005) Parameterized complexity of generalized vertex cover problems, algorithms and data structures, vol 3608. Springer, Berlin, pp 36–48

Karpinski M, Zelikovsky A (1997) Approximating dense cases of covering problems. In: Pardalos P, Du D (eds) Proceedings of the DIMACS workshop on network design: connectivity and facilities location. DIMACS series in discrete mathematics and theoretical computer science vol 40, pp 169C178

Li Y, Yang Z, Wang W (2017) Complexity and algorithms for the connected vertex cover problem in 4-regular graphs. Appl Math Comput 301:107–114

Mölle D, Richter S, Rossmanith P (2006) Enumerate and expand: new runtime bounds for vertex cover variants, computing and combinatorics, vol 4112. Springer, Berlin, pp 265–273

Mölle D, Richter S, Rossmanith P (2008) Enumerate and expand: improved algorithms for connected vertex cover and tree cover. Theor Comput Syst 43:234–253

Moser H (2005) Exact algorithms for generalizations of vertex cover, Masters thesis, Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena

Priyadarsini PK, Hemalatha T (2008) Connected vertex cover in 2-connected planar graph with maximum degree 4 is NP-complete. Int J Math Phys Eng Sci 2:51–54

Savage CD (1982) Depth-first search and the vertex cover problem. Inf Process Lett 14(5):233C237

Ueno S, Kajitani Y, Gotoh S (1988) On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. Discrete Math 72:355–360

Wanatabe T, Kajita S, Onaga K (1991) Vertex covers and connected vertex covers in 3-connected graphs. In: Circuits and systems ieee international sympoisum on, vol 2, pp 1017–1020

Zhang Z, Gao X, Wu W (2009) PTAS for connected vertex cover in unit disk graphs. Theor Comput Sci 410:5398–5402