Research Article

On the k-Component Independence Number of a Tree

Shuting Cheng and Baoyindureng Wu

Correspondence should be addressed to Baoyindureng Wu; baoyw@163.com

Received 11 January 2021; Accepted 27 May 2021; Published 7 June 2021

Copyright © 2021 Shuting Cheng and Baoyindureng Wu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Let $G$ be a graph and $k \geq 1$ be an integer. A subset $S$ of vertices in a graph $G$ is called a $k$-component independent set of $G$ if each component of $G[S]$ has order at most $k$. The $k$-component independence number, denoted by $\alpha^k_c(G)$, is the maximum order of a vertex subset that induces a subgraph with maximum component order at most $k$. We prove that if a tree $T$ is of order $n$, then $\alpha^1_c(T) \geq (k / (k + 1))n$. The bound is sharp. In addition, we give a linear-time algorithm for finding a maximum $k$-component independent set of a tree.

1. Introduction

Let $G = (V(G), E(G))$ be a graph and $k \geq 1$ be an integer and $S \subseteq V(G)$. We use $G[S]$ to denote the subgraph of $G$ induced by $S \subseteq V(G)$. We call $S$ a $k$-component independent set of $G$ if each component of $G[S]$ has order at most $k$. A $k$-component independent set is maximum if $G$ contains no larger $k$-component independent set and maximal if the set cannot be extended to a larger $k$-component independent set. The $k$-component independence number, denoted by $\alpha^k_c(G)$, is the cardinality of a maximum $k$-component independent set of $G$.

On the contrary, $S \subseteq V(G)$ is called a $k$-component vertex covering of $G$ if $V(G) \setminus S$ is a $k$-component independent set of $G$. A $k$-component vertex covering is minimum if $G$ contains no smaller $k$-component vertex covering and minimal if the set cannot be contained in a smaller $k$-component vertex covering. The $k$-component vertex covering number, denoted by $\beta^k_c(G)$, is the cardinality of a minimum $k$-component vertex covering of $G$.

By the definition above, for any graph $G$ of order $n$,

$$
\alpha^k_c(G) + \beta^k_c(G) = n,
$$

$$
\alpha^k_c(G) \geq \alpha^{(k-1)}_c(G) \geq \cdots \geq \alpha^1_c(G) = \alpha(G),
$$

where $\alpha(G)$ and $\beta(G)$ are the ordinary independence number and vertex covering number of $G$.

The $k$-component chromatic number of a graph $G$, denoted by $\chi^k_c(G)$, is the smallest number of colours needed in $k$-component coloring, a coloring of the vertices such that color classes are $k$-component independent sets. The notations $\alpha^k_c(G)$ and $\chi^k_c(G)$ come from [1]. The notion of $k$-component coloring is first studied by Kleinberg et al. [2]. It was extensively studied in the past two decades [3–9]. We refer to an excellent survey on this topic [10].

A notion, close to $k$-component vertex covering of a graph, is called the fragmentability of a graph, which was first introduced by Edwards and McDiarmid [11] when they were investigating the harmonious colorings of graphs. It was further studied in [12, 13].

Proposition 1. In general, deciding $\alpha^k_c(G)$ is NP-hard for a graph $G$.

Proof. Note that $(1/k)\alpha^k_c(G) \leq \alpha(G) \leq \alpha^k_c(G)$ for any graph $G$. If $\alpha^k_c(G)$ is determined by polynomial-time algorithm, then $\alpha(G)$ is determined by at most,

$$
\sum_{b=0}^{a} \binom{n}{b},
$$
additional check that whether \( S \) is an independent set or not, for every \( S \subseteq V(G) \) with \(|a| \leq |S| \leq a\), where \( a = d^*_G(G) \), contradicting the folklore that determining \( a(G) \) is NP-hard for a graph \( G \), in general.

In this note, we give a linear-time algorithm for finding a maximum \( k \)-component independence number of a tree.

2. An Lower Bound on \( a^*_c(T) \) for a Tree

Let \( G \) be a graph and \( x \in V(G) \). The order of \( G \) is denoted by \( V(G) \). We use \( N_G(x) \) denote the set of neighbors of a vertex \( x \) of \( G \). The degree of \( x \), denoted by \( d_G(x) \), is the number of edges incident with \( x \) in \( G \). Furthermore, the two symbols are simply denoted by \( N(x) \) and \( d(x) \), respectively. For a subset \( S \) of the vertex set \( V(G) \) of \( G \), \( G[S] \) denotes the subgraph of \( G \) induced by \( S \).

Let \( T \) be a tree with root \( r \). The level \( l(x) \) of a vertex \( x \) in \( T \) is the length of the path \( rTx \). Each vertex on the path \( rTx \) is called an ancestor of \( x \), and each vertex of, which \( x \) is an ancestor, is a descendant of \( x \). An ancestor or descendant of a vertex is proper if it is not the vertex itself. The immediate proper ancestor of a vertex \( x \in V(T) \backslash \{r\} \) is its predecessor or parent, denoted \( p(x) \). Let \( T_q \) denote the subtree of \( T \) with the vertex set which consists of the sets of descendants of \( x \).

**Lemma 1.** Let \( n \) and \( k \) be two integers with \( n \geq k + 1 \geq 2 \).

For any tree \( T \) of order \( n \), there exists a vertex \( v \) such that \( T - v \) has \( d(v) - 1 \) components, each of which has order at most \( k \), but the sum of their order is at least \( k \).

In particular, every nontrivial tree \( T \) has a vertex \( v \) such that all its neighbors but one are leaves.

**Proof.** Take a vertex \( r \in V(T) \) as the root of \( T \), thereby \( p(x) \) and \( l(x) \) of \( x \) are uniquely defined for each \( x \in V(T) \). Let \( q = \max\{l(x) \mid x \in V(T)\} \). Let \( V_q = \{x \mid x \in V(T) \mid l(x) = t \} \). Define a weight function \( w(u) = d(u) \) for each \( u \in V_q \). If there is a vertex \( u \in V_q \) such that \( w(u) \geq k \), then \( u \) is the vertex we desired.

Otherwise, \( w(u) \leq k - 1 \) for every vertex \( u \in V_q \). \( u \) follows that \( T - u \) has \( d(u) - 1 \) components, each of which has order at most \( k \) for each \( u \in V_q \). Define \( w(u) = \sum_{y \in u} w(y) \) for each \( x \in V_q \). If there is a vertex \( u \in V_q \) with \( w(u) \geq k \), then \( u \) is the vertex we desired. Otherwise, \( w(u) \leq k - 1 \) for every vertex \( u \in V_q \). \( u \) follows that \( T - u \) has \( d(u) - 1 \) components, each of which has order at most \( k \) for each \( u \in V_q \). Define a weight function \( w(x) = \sum_{y \in x} w(y) \) for each \( x \in V_q \). Repeat the procedure above; since \( n \geq k + 1 \) is finite, there exists an integer \( i \in \{0, 1, \ldots, q - 1\} \) such that there exists a vertex \( u_{q-i} \in V_{q-i} \) with \( w(u_{q-i}) \geq k \). It can be seen that \( u_{q-i} \) is the vertex we required.

**Theorem 1.** Let \( k \geq 1 \) be an integer. For any tree \( T \) of order \( n \), \( \beta^*_c(T) \leq (n/k + 1) \). Equivalently, \( a^*_c(T) \geq (k/(k + 1))n \). The bound is sharp.

**Proof.** We use induction on \( n \). If \( n \leq k \), then \( \beta^*_c(T) = 0 \), and the result trivially holds. Now, assume that \( n \geq k + 1 \). By Lemma 1, there exists a vertex \( v \) of \( T \) as the assertion in Lemma 1. Let \( T_1, \ldots, T_d \) be all components of \( T - v \) such that \( \sum_{i=1}^{d} v(T_i) \geq k \) and \( v(T) \leq k \), for each \( i \in \{1, \ldots, d - 1\} \), where \( d = d(v) \). By the induction hypothesis, \( \beta^*_c(T_d) \leq (v(T_d)/k + 1) \). So,

\[
\beta^*_c(k) \leq 1 + \beta^*_c(k) \leq 1 + \frac{v(T_d)}{k + 1} + \frac{n - k - 1}{k + 1} = \frac{n}{k + 1}.
\]

The bound is achieved by the path \( P_n \) of order \( n \) when \( n \) is divisible by \( k + 1 \).

By taking \( k = \lfloor n/2 \rfloor \) in the above theorem, we have the following.

**Corollary 1** (see [14]). For a tree \( T \) of order \( n \geq 2 \), \( \beta^*_c[T(n/2)] = 1 \); equivalently, there exists a vertex \( v \in V(T) \) such that each component of \( T - v \) has order at most \( n/2 \).

A path in a vertex-colored graph is called conflict-free if there is a color used on exactly one of its vertices. A vertex-colored graph is said to be conflict-free vertex-connected if any two vertices of the graph are connected by a conflict-free path. The conflict-free vertex-connection number, denoted by \( vfc(G) \), is defined as the smallest number of colours required to make \( G \) conflict-free vertex connected. Li et al. [15] conjectured that, for a connected graph \( G \) of order \( n \), \( vfc(G) \leq vfc(P_n) \). Using Corollary 1, the authors of [14] are able to confirm the above conjecture. We refer to [16–18], for more recent results, on conflict-free vertex-connection of graphs. Next we give a linear time algorithm (Algorithm 1) for finding minimum \( k \)-component vertex covering of a tree.

3. Linear-Time Algorithm

**Theorem 2.** Every \( C \) returned by the algorithm is a minimum \( k \)-component vertex covering of \( T \).

**Proof.** We prove it by the induction on \( v(T) \). If \( v(T) \leq k \), \( C \) returned by the algorithm is the empty set and thus is a minimum \( k \)-component vertex covering of \( T \) since \( \beta^*_c(T) = 0 \).

Next, assume that \( v(G) \geq k + 1 \). Let \( C = \{v_1, v_2, \ldots, v_l\} \), where \( v_1 \) is the first vertex added to \( C \) by the algorithm. By the choice of the algorithm, \( v_1 \) is a vector with the property described in the assertion in Lemma 1. Let \( T_1, \ldots, T_{d-1}, T_d \) be all components of \( T - v_1 \) such that \( \sum_{i=1}^{d} v(T_i) \geq k \) and \( v(T) \leq k \), for each \( i \in \{1, \ldots, d - 1\} \), where \( d = d(v_1) \). By the induction hypothesis, \( \{v_2, \ldots, v_l\} \) is a minimum \( k \)-component vertex covering of \( T_d \). Thus, \( C \) is a \( k \)-component vertex covering of \( T \).

Suppose \( C \) is not a minimum \( k \)-component vertex covering of \( T \), and let \( C^* \) be a minimum \( k \)-component vertex covering of \( T \). It is clear that \( v_1 \notin C^* \). Note that \( C^* = (C^* \cap V(T_d)) \cup \{v_1\} \) is a \( k \)-component vertex covering of \( T \). Thus, \( |C^* \cap V(T_d)| \geq |\{v_2, \ldots, v_l\}| = |C| - 1 \).

We prove it by the induction on \( |C| \).
Input: a tree $T$ with a vertex $r$ as its root.
Output: a minimum $k$-component vertex covering $C$ of $T$.

\begin{enumerate}
\item $C_0 = \emptyset$
\item while $v(T) > k$, do
\item set $q_0 = \max\{l(x) | x \in V(T)\}$, $S_0 = \{x | l(x) = q_0 - 1\}$, $w_0(x) = d(x) - 1$ for each $x \in S_0$
\item if $S_0 \neq \emptyset$
\item choose a vertex $x \in S_0$
\item if $w(x) \geq k$, set $C = C_0 \cup \{x\}$, $T = T - T_x$ go to Step 2
\item if $w(x) < k$, put $w(x) = w(x) + 1$, and put $S_0 = S_0 \setminus \{x\}$, go to Step 4
\item else
\item replace $S_0 = \sum_{y \in S} p(y)$
\item $w(x) = \sum_{p(y) \leq x} w(y)$ for each $x \in S_0$, go to Step 4
\item end if
\item end while
\item end while
\end{enumerate}

(13) return $C$

Algorithm 1: Algorithm for finding minimum $k$-component vertex covering.

Since $\sum_{i=1}^{n-1} v(T_i) \geq k$, $C^* \cap (\cup_{i=1}^{n-1} T_i) \neq \emptyset$. It follows that $|C^*| \geq 1 + |C^* \cap V(T_n)| \geq t = |C|$, a contradiction. \hfill $\square$

In the execution of the algorithm, each vertex $x$ is explored at most once to check whether $w(x) \geq k$ or not. So, the running time of the algorithm is $O(n)$.

4. Further Research

Zito [19] determined that the greatest number of maximum independent sets of a tree of order $n$ is,

$$\begin{cases} 2^{(n-3)/2}, & \text{if } n > 1 \text{ is odd}, \\ 2^{(n-2)/2} + 1, & \text{if } n \text{ is even}. \end{cases}$$

(4)

More relevant work can be found in [20–22]. Naturally, one asks the following questions:

1. What is the largest number of maximum (or maximal) $k$-component independent sets on a tree of order $n$?
2. What is the largest number of maximum (or maximal) $k$-component independent sets on a (or connected) graph of order $n$?
3. What is the largest number of maximum (or maximal) $k$-component independent sets on a (or connected) graph of order $n$ vertices and $m$ edges?

Data Availability

No data were used to support the findings of the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

All authors contributed equally and significantly in conducting this research work and writing this paper.

Acknowledgments

This research was supported by the Key Laboratory Project of Xinjiang (2018D04017), NSFC (no. 12061073), and Xinjiang Education Department XJEDU2019I001.

References

[1] N. Broutin and R. J. Kang, “Bounded monochromatic components for random graphs,” Journal of Combinatorics, vol. 9, no. 3, pp. 411–446, 2018.
[2] J. Kleinberg, R. Motwani, P. Raghavan, and S. Venkatasubramanian, “Storage management for evolving databases,” in Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS 97), pp. 353–362, Miami Beach, FL, USA, October 1997.
[3] N. Alon, G. Ding, B. Oporowski, and D. Vertigan, “Partitioning into graphs with only small components,” Journal of Combinatorial Theory, Series B, vol. 87, no. 2, pp. 231–243, 2003.
[4] G. Chappell and J. Gimbel, “On subgraphs without large components,” Mathematica Bohemica, vol. 142, no. 1, pp. 85–111, 2017.
[5] I. Choi, F. Dross, and P. Ochem, “Partitioning sparse graphs into an independent set and a graph with bounded size components,” Discrete Mathematics, vol. 343, no. 8, Article ID 111921, 2020.
[6] L. Esperet and P. Ochem, “Islands in graphs on surfaces,” SIAM Journal on Discrete Mathematics, vol. 30, no. 1, pp. 206–219, 2016.
[7] P. Hazell, T. Szabó, and G. Tardos, “Bounded size components-partitions and transversals,” Journal of Combinatorial Theory, Series B, vol. 88, no. 2, pp. 281–297, 2003.
[8] N. Linial, J. Matousek, O. Sheffet, and G. Tardos, “Graph colouring with no large monochromatic components,” Combinatorics, Probability and Computing, vol. 17, no. 4, pp. 577–589, 2008.
[9] S. Norin, A. Scott, P. Seymour, and D. R. Wood, “Clustered colouring in minor-closed classes,” Combinatorica, vol. 39, no. 6, pp. 1387–1412, 2019.
[10] D. R. Wood, “Defective and clustered graph colouring,” Electronic Journal of Combinatorics, vol. 13, p. DS23, 2018.
[11] K. Edwards and C. McDiarmid, “New upper bounds on harmonious colorings,” Journal of Graph Theory, vol. 18, no. 3, pp. 257–267, 1994.

[12] K. Edwards and G. Farr, “Fragmentability of graphs,” Journal of Combinatorial Theory, Series B, vol. 82, no. 1, pp. 30–37, 2001.

[13] K. J. Edwards and G. E. Farr, “Graph fragmentability,” in Topics in Structural Graph Theory, Encyclopedia of Mathematics and its Applications No. 147, L. W. Beineke and R. J. Wilson, Eds., pp. 203–218, Cambridge University Press, Cambridge, UK, 2013.

[14] Z. Li and B. Wu, “Maximum value of conflict-free vertex-connection number of graphs,” Discrete Mathematics, Algorithms and Applications, vol. 10, no. 5, Article ID 1850059, 2018.

[15] X. Li, Y. Zhang, X. Zhu, Y. Mao, and H. Zhao, “Conflict-free vertex-connections of graphs,” http://arxiv.org/abs/705.07270v1.

[16] H. Chang, Z. Huang, X. Li, Y. Mao, and H. Zhao, “On conflict-free connection of graphs,” Discrete Applied Mathematics, vol. 255, pp. 167–182, 2019.

[17] X. Li and X. Zhu, “Conflict-free (vertex-)connection numbers of graphs with small diameter,” Australasian Journal of Combinatorics, vol. 76, no. 1, pp. 288–298, 2020.

[18] J. Meng, X. Li, and X. Zhu, “(Strong) conflict-free connectivity: algorithm and complexity,” Theoretical Computer Science, vol. 804, pp. 72–80, 2020.

[19] J. Zito, “The structure and maximum number of maximum independent sets in trees,” Journal of Graph Theory, vol. 15, no. 2, pp. 207–221, 1991.

[20] Z. Jin and X. Li, “Graphs with the second largest number of maximal independent sets,” Discrete Mathematics, vol. 308, no. 23, pp. 5864–5870, 2008.

[21] M. Jou and G. Chang, “The number of maximum independent sets in graphs, Taiwan,” Journal of Mathematics, vol. 4, pp. 685–695, 2000.

[22] E. Mohr and D. Rautenbach, “On the maximum number of maximum independent sets,” Graphs and Combinatorics, vol. 34, no. 6, pp. 1729–1740, 2018.