A polynomial-time algorithm of finding a minimum k-path vertex cover and a maximum k-path packing in some graphs

D. B. Mokeev & D. S. Malyshev

Optimization Letters

ISSN 1862-4472
Volume 14
Number 6

Optim Lett (2020) 14:1317-1322
DOI 10.1007/s11590-019-01475-0

Springer

Optimization Letters (2020) 14:1317–1322
https://doi.org/10.1007/s11590-019-01475-0

ORIGINAL PAPER

A polynomial-time algorithm of finding a minimum *k*-path vertex cover and a maximum *k*-path packing in some graphs

D. B. Mokeev[1,2] · D. S. Malyshev[3]

Received: 20 June 2019 / Accepted: 29 August 2019 / Published online: 3 September 2019

Abstract

For a graph G and a positive integer k, a subset C of vertices of G is called a k-path vertex cover if C intersects all paths of k vertices in G. The cardinality of a minimum k-path vertex cover is denoted by $\beta_{P_k}(G)$. For a graph G and a positive integer k, a subset M of pairwise vertex-disjoint paths of k vertices in G is called a k-path packing. The cardinality of a maximum k-path packing is denoted by $\mu_{P_k}(G)$. In this paper, we describe some graphs, having equal values of β_{P_k} and μ_{P_k}, for $k \geq 5$, and present polynomial-time algorithms of finding a minimum k-path vertex cover and a maximum k-path packing in such graphs.

Keywords k-path vertex cover · k-path packing · Computational complexity

1 Introduction

By default, all graphs in this paper are finite, undirected, without loops and multiple edges. We use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of a graph G, respectively. We call a *k-path* a path of k vertices and use P_k to denote it.

For a positive integer k, a set of pairwise vertex-disjoint k-paths of a graph G is called a *k-path packing* of G. The *k-path packing problem* is to find a maximum

✉ D. S. Malyshev
dsmalyshev@rambler.ru; dmalishev@hse.ru

D. B. Mokeev
mokeevdb@gmail.com

1 National Research Lobachevsky State University of Nizhni Novgorod, 23 Gagarina Ave., Nizhny Novgorod, Russia 603950

2 Laboratory of Algorithms and Technologies for Networks Analysis, National Research University Higher School of Economics, 136 Rodionova Str., Nizhny Novgorod, Russia 603093

3 National Research University Higher School of Economics, 25/12 Bolshaja Pecherskaja Ulitsa, Nizhny Novgorod, Russia 603155

k-path packing in a graph. For a positive integer k, a set of vertices of a graph G, which intersects all k-paths of G, is called a *k-path vertex cover* of G. The *k-path vertex cover problem* is to find a minimum k-path vertex cover in a given graph. For a given graph H, the H-packing problem can be defined in a similar way. The k-path vertex cover problem can be motivated by a problem, related to security protocols in wireless sensor networks (see, for example [2,10,12,14]) or the problem of installing cameras on roads [11].

A lot of papers on packing problems are devoted to algorithmic aspects (see [4,6, 7,13]). It is known that the matching problem (i.e., the 2-path packing problem) can be solved in polynomial time [3], but the H-packing problem is NP-complete for any graph H, having a connected component on three or more vertices [5].

It seems perspective to find new polynomially solvable cases for the k-path packing and k-path vertex cover problems. Several results are known for $k = 3$ [1], $k = 4$ [9], and for the general case [8].

The aim of this paper is to describe a family of graph classes, on which the k-path packing and k-path vertex cover problems for $k \geq 5$ have polynomial-time algorithms. Namely, we consider some graphs, which hereditarily meet the equality of β_{P_k} and μ_{P_k}. This property admits us to present a polynomial-time algorithm of finding a minimum k-path vertex cover and a maximum k-path packing in such graphs.

We denote by $(v_1, v_2, \ldots, v_k)$ a k-path that consists of vertices $v_1, v_2, \ldots, v_k$. We denote by $|G|$ the number of vertices in G. We denote by $G \cup H$ the graph, obtained from graphs G and H by their union.

For a given graph G and its subgraph H, we denote by $G \backslash H$ the graph, obtained from G by deleting each vertex of H with all incident edges. For a given graph G and $A \subseteq V(G)$, we denote by $G[A]$ the subgraph of G, induced by the set A.

2 *k*-extended graphs

In this section, we describe k-extended graphs and prove the equality of β_{P_k} and μ_{P_k} for such graphs.

Definition 1 An induced subgraph T of a graph G is a *terminal subgraph* of G if there is only one vertex u of the graph $G \backslash T$, which is adjacent to one or more vertices of T. We call u the *contact vertex* of T.

For any $k \geq 2$, we call a connected graph, which does not have a k-path, as a F_k-graph.

Definition 2 Let $\mathcal{M}$ be a pseudograph (a graph with possible loops and multiple edges). Given an integer $k \geq 5$, the operation of a *k-extension* of $\mathcal{M}$ consists of the following. All edges of cycles, including loops and two or more edges between the same vertices, are subdivided, each with $k - 1$ vertices. For a vertex v, denote by $d(v)$ the distance between v and a nearest vertex of $\mathcal{M}$. For each new vertex x with $d(x) \geq 2$, several terminal $F_{d(x)}$-graphs with a contact vertex x can be added. For each old vertex y, several terminal F_k-graphs with a contact vertex y can be added.

We refer to the obtained graph as k-extended graph.

Theorem 1 *For each integer $k \geq 5$, $\beta_{P_k}(G) = \mu_{P_k}(G)$ in every k-extended graph G.*

Proof Let G be obtained by the operation of a k-extention from a pseudograph $\mathcal{M}$.

We can assume that G is a connected graph. Otherwise, we can consider any its connected component. The proof is by induction on the number of vertices in G. If G does not have k-paths, then $\mu_{P_k}(G) = \beta_{P_k}(G) = 0$.

Let $\mu_{P_k}(G) \geq 1$ and $\beta_{P_k}(H) = \mu_{P_k}(H)$, for each graph H with $|H| < |G|$.

Consider the following cases.

1. There exists a terminal F_k-subgraph X with the contact vertex a in G, such that $G[V(X) \cup \{a\}]$ contains a k-path. Or there exists terminal F_k subgraphs X_1 and X_2 with a common contact vertex a in G, such that none of $G[V(X_1) \cup \{a\}]$ and $G[V(X_2) \cup \{a\}]$ contains k-paths, but $G[V(X) \cup V(Y) \cup \{a\}]$ contains a k-path. Denote $X = X_1 \cup X_2$ in the second case. One can see that each k-path, which contains vertices of the set $V(X)$, contains also the vertex a.
   Consider an arbitrary k-path P of $G[V(X) \cup \{a\}]$. Denote $G^{'} = G \backslash P$. By the induction hypothesis, there exist a k-path packing M and a k-path vertex cover C of $G^{'}$, such that $|M| = |C|$.
   Then $M \cup \{P\}$ is the k-path packing of G of the cardinality $|M| + 1$ and $C \cup \{a\}$ is the k-path vertex cover of G of the same cardinality. Therefore, $\mu_{P_k}(G) = \beta_{P_k}(G)$.
2. The graph G has not terminal F_k-subgraphs with the properties above. Then G contains a cycle of nk vertices, where $n \in \mathbb{N}$. Consider a cycle Y_0 of $\mathcal{M}$. Denote by Y_1 a cycle of G, which is obtained from Y_0 by $k - 1$ vertex subdivisions of all its edges.
   Denote $G^{'} = G \backslash Y_1$. By the induction hypothesis, there exist a k-path packing M and a k-path vertex cover C of $G^{'}$, such that $|M| = |C|$.
   Denote $t = |Y_0|$. Then $|Y_1| = kt$, i.e. the cycle Y_1 can be split into t pairwise vertex-disjoint k-paths. Denote such k-paths as $P_1, P_2, \ldots, P_t$. Then $M \cup \{P_1, P_2, \ldots, P_t\}$ is a k-path packing of G of the cardinality $|M| + t$.
   We need to prove that $C \cup V(Y_0)$ is a k-path vertex cover of G. Note, there are not edges between the vertex sets $V(G \backslash Y_2)$ and $V(Y_2 \backslash Y_0)$. So, each path, containing vertices of the both sets, contains at least one vertex of Y_0.
   Consider a connected component Z of the graph $Y_2 \backslash Y_0$. It consists of a $(k-1)$-path P and some terminal subgraphs with the contact vertices from P. Let x be a vertex of P, and T be a terminal subgraph with the contact vertex x. One can see that $d(x)$ equals the difference between the radius of P and its distance from the center of P. Since T is a $F_{d(x)}$-graph, the graph $G[V(T) \cup \{x\}]$ has not $(d(x)+1)$-paths. Thus, none of the paths in Z has length more than $k - 1$, i.e. Z is the F_k-graph. Hence, each connected component of the graph $Y_2 \backslash Y_0$ is a F_k-graph.
   Hence, each k-path of Y_2 contains at least one vertex of Y_0. So, $C \cup V(Y_0)$ is a k-path vertex cover of G of the cardinality $|C| + t = |M| + t$.
   Therefore, $\mu_{P_k}(G) = \beta_{P_k}(G)$.

□

3 Algorithms

Here we show that we can find a maximum k-path packing and a minimum k-path vertex cover in k-extended graphs in $O(n^2)$ time, where n is the number of vertices in an input graph.

Let $\mathcal{M}(G)$ denote a pseudograph, such that G is obtained by the operation of a k-extention from $\mathcal{M}(G)$. Let A be the set of all cyclic vertices of $\mathcal{M}(G)$. The set A can be found in $O(n^2)$ time, using the depth-first search (see Algorithm 1).

Algorithm 1.
Input: A k-extended connected graph $G = (V, E)$ with $|V| \geq k$.
Output: The set A of all cyclic vertices of $\mathcal{M}(G)$.

1. $A = \emptyset$; $B = \emptyset$.
2. Choose an arbitrary vertex $z \in V$.
3. Build a DFS-tree T of G with the root z.
4. $E^{'} = E \backslash E(T)$.
5. **For** each $e \in E^{'}$ **do**
6.   Find a cycle C in the graph $(V(T), E(T) \cup \{e\})$.
7.   **If** $|C| \geq k$, **then** add $V(C)$ into B **End If**
8. **End For**
9. **For** each $v \in B$ **do**
10.   **If** $deg(v) \geq 2$ in $G[B]$, **then**
11.   add v into A.
12.   **For** each $u \in B$ **do**
13.   **If** $dist(v, u)$ is divisible by k, **then** add u into A **End If**
14.   **End For**
15.   **End If**
16. **End For**

From the proof of Theorem 1, we can see that the following algorithm finds a maximum k-path packing and a minimum k-path vertex cover in a connected k-extended graph G.

Algorithm 2.
Input: A k-extended connected graph $G = (V, E)$ with $|V| \geq k$.
Output: A vertex set $C \subseteq V$, which is a minimum k-path vertex cover of G; a k-path set M, which is a maximum k-path packing of G.

1. $C = \emptyset$; $M = \emptyset$.
2. **For** each vertex $v \in V$ **do** $l(v) = 0$ **End For**
3. Find the set A.
4. Choose $z \in V$. **If** $A = B = \emptyset$, **then** z is an arbitrary leaf of G, **else** z is an arbitrary vertex of A.
5. Build a DFS-tree T of G with the root z.
   Denote by $p(v)$ the parent of the vertex v in T.
   Denote by $Ch(v)$ the set of all children of the vertex v in T.
6. **For** each leaf v of T **do** $l(v) = 1$ **End For**
7. **While** $|V(T)| \geq k$ **do**

Choose v, such that $l(v) = 0$ and $l(u) > 0$, for each $u \in Ch(v)$.
Choose $x \in Ch(v)$, where $l(x) \geq l(u)$, for each $u \in Ch(v)$.
If $l(x) = k - 1$, **then**
  Add v into C.
  Find a k-path P in the subtree with the root v.
  Add P into M.
  Delete the subtree with the root v from T.
  $l(p(v)) = 1$.
Else
  Choose $y \in Ch(v) \setminus \{x\}$, where $l(y) \geq l(u)$, for each $u \in Ch(v) \setminus \{x\}$.
  If $l(x) + l(y) \geq k - 1$, **then**
    Add v into C.
    Find a k-path P in the subtree with the root v.
    Add P into M.
    Delete the subtree with the root v from T.
    $l(p(v)) = 1$.
  Else
    $l(k) = l(x) + 1$.
  End If
End If
End While

Note that the complexity of building a DFS-tree for a graph is $O(n^2)$ and there is only one cycle in the other part of the algorithm. So, for any fixed k, the complexity of Algorithm 2 is $O(n^2)$. If the graph G is not connected, then we can repeat this algorithm for each its connected component. Hence, a maximum k-path packing and a minimum k-path vertex cover can be found in k-extended graphs in time $O(n^2)$.

Acknowledgements The results of Sect. 2 were prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE). The results of Sect. 3 were obtained with the support of the RFFI Grant 18-31-20001-mol-a-ved.

References

1. Alekseev, V.E., Mokeev, D.B.: König graphs for 3-paths and 3-cycles. Discrete Appl. Math. **204**, 1–5 (2016)
2. Brešar, B., Kardoš-Belluš, R., Semanišin, G., Šparl, P.: On the weighted k-path vertex cover problem. Discret. Appl. Math. **177**, 14–18 (2014)
3. Edmonds, J.: Paths, trees, and flowers. Can. J. Math. **17**, 449–467 (1965)
4. Hell, P.: Graph packing. Electron. Notes. Discrete Math. **5**, 170–173 (2000)
5. Kirkpatrick, D. G., Hell, P.: On the completeness of a generalized matching problem. In: Proceedings of the Tenth Annual ACM Symposium on Theory of computing (San Diego, May 1–3). ACM, New York, pp. 240–245 (1978)
6. Malyshev, D.S.: Boundary graph classes for some maximum induced subgraph problems. J. Comb. Optim. **27**, 345–354 (2014)
7. Malyshev, D.S.: The impact of the growth rate of the packing number of graphs on the computational complexity of the independent set problem. Discrete Math. Appl. **23**, 245–249 (2013)
8. Masuyama, S., Ibaraki, T.: Chain packing in graphs. Algorithmica **6**, 826–839 (1991)
9. Mokeev, D.B.: On König graphs with respect to P_4. J. Appl. Ind. Math. **11**, 421–430 (2017)

10. Novotny, M.: Formal analysis of security protocols for wireless sensor networks. Tatra Mt. Math. Publ. **47**, 81–97 (2010)
11. Tu, J., Zhou, W.: A primal-dual approximation algorithm for the vertex cover P_3 problem. Theor. Comput. Sci. **412**, 7044–7048 (2011)
12. Yannakakis, M.: Node-deletion problems on bipartite graphs. SIAM J. Comput. **10**, 310–327 (1981)
13. Yuster, R.: Combinatorial and computational aspects of graph packing and graph decomposition. Comput. Sci. Rev. **1**, 12–26 (2007)
14. Zhang, B., Zuo, L.: The k-path vertex cover of some product graphs. WSEAS Trans. Math. **15**, 374–384 (2016)