ELSEVIER

Discrete Applied Mathematics 78 (1997) 227–233

DISCRETE APPLIED MATHEMATICS

k-Path partitions in trees

Jing-Ho Yan [a,1], Gerard J. Chang [a,1], Sandra M. Hedetniemi [b,2], Stephen T. Hedetniemi [b,*,2]

[a] *Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan*
[b] *Department of Computer Science, Clemson University, Clemson, SC 29634-1906, USA*

Received 15 August 1995; revised 19 November 1996

Abstract

For a fixed positive integer k, the k-path partition problem is to partition the vertex set of a graph into the smallest number of paths such that each path has at most k vertices. The 2-path partition problem is equivalent to the edge-cover problem. This paper presents a linear-time algorithm for the k-path partition problem in trees. The algorithm is applicable to the problem of finding the minimum number of message originators necessary to broadcast a message to all vertices in a tree network in one or two time units.

1. Introduction

A *path partition* of a graph G is a collection of vertex disjoint paths whose union is $V(G)$. The *path partition problem* is the problem of determining the minimum number of paths $p(G)$ in a path partition of G. Note that a graph G has a Hamiltonian path if and only if $p(G)=1$. Since the Hamiltonian path problem is NP-complete for planar graphs, bipartite graphs and chordal graphs (see [5]), so is the path partition problem. Bonuccelli and Bovet [3] and Arikati and Pandu Rangan [2] gave linear-time algorithms for the path partition problem in interval and circular arc graphs, Goodman and Hedetniemi [6] and Misra and Tarjan [9] gave a linear-time algorithms for trees, Skupien [10] gave a polynomial algorithm for forests, Chang and Kuo [4] gave a linear-time algorithm for cographs, and Srikant et al. [11] gave linear-time algorithms for bipartite permutation graphs and block graphs. In fact, Srikant et al.'s algorithm does not work for all block graphs. Yan and Chang [13] gave a linear-time algorithm for block graphs.

A generalization of the path partition problem is as follows. For a fixed positive integer k, a path partition is called a *k-path partition* if each of its paths has at

* Corresponding author. E-mail: hedet@cs.clemson.edu.

[1] Supported in part by the National Science Council under grant NSC83-0208-M009-050.
[2] Supported in part by the National Science Foundation under grant ENG-79-02960.

PII S0166-218X(97)00012-7

most k vertices. The *k-path partition problem* is the problem of determining the *k-path number* $p_k(G)$, which equals the minimum cardinality of a k-path partition of G. The 2-path partition problem is equivalent to the *edge-cover problem*, which is the problem of determining the minimum number of edges and isolated vertices which contain all vertices. Note that the n-path partition problem is the same as the path partition problem in a graph of n vertices.

The k-path partition problem is applicable to the following broadcasting problem. In computer or communication networks there frequently arises a situation where some information must be communicated from some vertices to all other vertices in the network. We refer to this as *broadcasting*. For a good survey, see [7]. In this paper we are concerned with the problem of determining the minimum number of message originators necessary to complete broadcasting within a fixed number of time units. More precisely, we model a communication network with a graph $G=(V,E)$, where the edges E represent the communication lines of the network. All communication is done by placing phone calls over the edges of G subject to the following restrictions:

(1) a vertex may participate in only one call per unit of time;

(2) a vertex may only call an adjacent vertex; and

(3) each call requires one unit of time to communicate the information.

It is easy to see that for any connected graph G, the minimum number of vertices from which broadcasting can be completed in two (resp. one) time units equals $p_4(G)$ (resp. $p_2(G)$).

A slightly more general version of the k-path *partition problem* has recently been studied by Abbas [1]. In her Ph.D. thesis, Abbas studied a variety of graph clustering problems, including the problem of partitioning a graph into a minimum number of subgraphs of bounded diameter. Abbas showed that this problem is NP-complete on bipartite and chordal graphs, and gave linear time sequential algorithms for this problem on bipartite permutation and interval graphs.

The purpose of this paper is to present a linear-time algorithm for the k-path partition problem in trees. By the above discussion, this algorithm can be used to find the minimum number of message originators necessary to broadcast a message to all vertices in a tree network in one or two time units. For technical reasons, we also consider the k-path partition problem with an additional condition. Suppose v is a fixed vertex of a graph G. We consider G to be a graph "rooted" at v. A *rooted k-path partition* of the graph G is a k-path partition in which v is an endvertex of a path in the partition. The *rooted k-path partition number* $p_k(G,v)$ is the minimum cardinality of a rooted k-path partition of G. Furthermore, let $l_k(G,v)$ denote the minimum number of vertices in a path containing v in a rooted k-path partition of size $p_k(G,v)$. In this paper, we give recursive formulas for $p_k(G), p_k(G,v)$ and $l_k(G,v)$ in terms of $p_k(G_i)$'s, $p_k(G_i,v_i)$'s and $l_k(G_i,v_i)$'s, where the G_i's are subtrees of a larger tree G. From this, we obtain a linear-time algorithm for the k-path partition problem in trees.

Recall that a tree is an acyclic, connected graph. A fundamental property useful to our discussion is that a non-trivial tree has at least two *leaves*, i.e. vertices of degree one. Conversely, trees can be obtained from a trivial graph by repeatedly adding new vertices and joining them to existing vertices. An alternative description is by means of the following composition operation: Suppose G_1 and G_2 are two disjoint graphs rooted at v_1 and v_2, respectively. The *composition* of G_1 and G_2 is the graph G rooted at v_1 that is obtained from the disjoint union of G_1 and G_2 by adding a new edge v_1v_2. Note that any tree can be obtained from trivial graphs by a sequence of graph compositions.

2. Main theorem

This section establishes some basic theorems for designing a linear-time algorithm for the k-path partition problem in trees. Note that the lemmas and theorems established in this section apply to arbitrary graphs, even though we apply them only to trees.

Suppose P is a k-path partition of a graph G. For any induced subgraph H of G, let P_H denote the k-path partition of H resulting from P when each vertex in $G - H$ is deleted from the path containing it in P.

Lemma 1. $p_k(G,v) - 1 \leqslant p_k(G) \leqslant p_k(G,v)$ *for a graph G with root v.*

Proof. Since a rooted k-path partition is a k-path partition, we have $p_k(G) \leqslant p_k(G,v)$. Suppose P is an optimal k-path partition of G and p is the path containing v in P. We can partition p into two k-paths p_1 and p_2 such that v is an endvertex of p_1 and p_2 may be empty. Then $P - \{p\} \cup \{p_1, p_2\}$ is a rooted k-path partition of G of size at most $p_k(G) + 1$. Hence, $p_k(G,v) - 1 \leqslant p_k(G)$. □

Lemma 2. *If G is the composition of two graphs G_1 and G_2 with roots v_1 and v_2 respectively, then* (a) *and* (b) *hold.*

(a) $p_k(G_1) + p_k(G_2) - 1 \leqslant p_k(G) \leqslant p_k(G_1) + p_k(G_2)$.

(b) $p_k(G_1, v_1) + p_k(G_2) - 1 \leqslant p_k(G, v_1) \leqslant p_k(G_1, v_1) + p_k(G_2)$.

Proof. (a) Suppose P is an optimal k-path partition of G. Then P_{G_1} and P_{G_2} are k-path partitions of G_1 and G_2, respectively, and

$$p_k(G) \geqslant |P_{G_1}| + |P_{G_2}| - 1 \geqslant p_k(G_1) + p_k(G_2) - 1.$$

On the other hand, suppose P_i is an optimal k-path partition of G_i for $i = 1, 2$. Then $P_1 \cup P_2$ is a k-path partition of G and $p_k(G) \leqslant p_k(G_1) + p_k(G_2)$.

(b) Suppose P is an optimal rooted k-path partition of G. Then P_{G_1} is a rooted k-path partition of G_1, P_{G_2} a k-path partition of G_2 and

$$p_k(G, v_1) \geqslant |P_{G_1}| + |P_{G_2}| - 1 \geqslant p_k(G_1, v_1) + p_k(G_2) - 1.$$

Conversely, suppose P_1 is an optimal rooted k-path partition and P_2 an optimal k-path partition. Then $P_1 \cup P_2$ is a rooted k-path partition of G and $p_k(G, v_1) \leqslant p_k(G_1, v_1) + p_k(G_2)$. □

Theorem 3. *If G is the composition of two graphs G_1 and G_2 with roots v_1 and v_2 respectively, then*

$$p_k(G)=\begin{cases} p_k(G_1)+p_k(G_2)-1 & \text{if } |A|=2 \text{ and } l_k(G_1,v_1)+l_k(G_2,v_2)\leqslant k,\\ p_k(G_1)+p_k(G_2) & \text{otherwise,}\end{cases}$$

where

$$A=\{i\mid\ p_k(G_i)=p_k(G_i,v_i),\ i=1 \text{ or } 2\}.$$

Proof. By Lemma 2(a), $p_k(G_1)+p_k(G_2)-1\leqslant p_k(G)\leqslant p_k(G_1)+p_k(G_2)$. So we need to show only that $p_k(G)=p_k(G_1)+p_k(G_2)-1$ if and only if $|A|\leqslant 2$ and $l_k(G_1,v_1)+l_k(G_2,v_2)\leqslant k$.

Suppose $p_k(G)=p_k(G_1)+p_k(G_2)-1$. Let P be an optimal k-path partition of G and q be the path containing v_1 in P. Then P_{G_1} and P_{G_2} are k-path partitions of G_1 and G_2, respectively. If $q\cap V(G_2)=\emptyset$, then

$$|P_{G_1}|+|P_{G_2}|=p_k(G)=p_k(G_1)+p_k(G_2)-1.$$

This implies $|P_{G_1}|<p_k(G_1)$ or $|P_{G_2}|<p_k(G_2)$, a contradiction. Therefore $q\cap V(G_2)\neq\emptyset$. It follows that P_{G_i} is a rooted k-path partition of G_i for $i=1,2$. Since $|P_{G_1}|+|P_{G_2}|=p_k(G)+1=p_k(G_1)+p_k(G_2)$, $p_k(G_i)\leqslant p_k(G_i,v_i)\leqslant|P_{G_i}|=p_k(G_i)$ for $i=1,2$. So $p_k(G_i)$
$=p_k(G_i,v_i)$ and P_{G_i} is an optimal rooted k-path partition for $i=1,2$. This implies that $|A|=2$ and $l_k(G_1,v_1)+l_k(G_2,v_2)\leqslant|q|\leqslant k$.

On the other hand, suppose $|A|=2$ and $l_k(G_1,v_1)+l_k(G_2,v_2)\leqslant k$. Let P_i be an optimal rooted k-path partition of G_i and q_i be the path containing v_i in P_i for $i=1,2$. Since $|q_1+q_2|=l_k(G_1,v_1)+l_k(G_2,v_2)\leqslant k$, $(P_1\cup P_2\cup\{q_1\cup\{v_1v_2\}\cup q_2\})-\{q_1,q_2\}$ is a k-path partition of G of size $p_k(G_1,v_1)+p_k(G_2,v_2)-1=p_k(G_1)+p_k(G_2)-1$ and so $p_k(G)=p_k(G_1)+p_k(G_2)-1$. □

Theorem 4. *If G is the composition of two graphs G_1 and G_2 with roots v_1 and v_2, respectively, then*

$$p_k(G,v_1)=\begin{cases} p_k(G_1,v_1)+p_k(G_2)-1 & \text{if } l_k(G_1,v_1)=1 \text{ and } l_k(G_2,v_2)<k\\ & \text{and } p_k(G_2)=p_k(G_2,v_2),\\ p_k(G_1,v_1)+p_k(G_2) & \text{otherwise.}\end{cases}$$

Proof. By Lemma 2(b), $p_k(G_1,v_1)+p_k(G_2)-1\leqslant p_k(G,v_1)\leqslant p_k(G_1,v_1)+p_k(G_2)$. So we need to show only that $p_k(G,v_1)=p_k(G_1,v_1)+p_k(G_2)-1$ if and only if $l_k(G_1,v_1)=1$ and $l_k(G_2,v_2)<k$ and $p_k(G_2)=p_k(G_2,v_2)$.

Suppose $p_k(G,v_1)=p_k(G_1,v_1)+p_k(G_2,v_2)-1$. Let P be an optimal rooted k-path partition of G and q be the path containing v_1 in P. Then P_{G_1} is a rooted k-path partition of G_1 and P_{G_2} is a k-path partition of G_2. If $q\cap V(G_2)=\emptyset$, then

$$|P_{G_1}|+|P_{G_2}|=p_k(G,v_1)=p_k(G_1,v_1)+p_k(G_2)-1.$$

This implies $|P_{G_1}| < p_k(G_1, v_1)$ or $|P_{G_2}| < p_k(G_2)$, a contradiction. Therefore $q \cap V(G_2) \neq \emptyset$ and $q \cap V(G_1) = \{v_1\}$. It follows that P_{G_2} is also a rooted k-path partition of G_2. Since $|P_{G_1}| + |P_{G_2}| = p_k(G, v_1) + 1 = p_k(G_1, v_1) + p_k(G_2)$, $|P_{G_1}| = p_k(G_1, v_1)$ and $p_k(G_2) \leqslant p_k(G_2, v_2) \leqslant |P_{G_2}| = p_k(G_2)$. So $p_k(G_2) = p_k(G_2, v_2)$ and P_{G_i} is an optimal rooted k-path partition for $i = 1, 2$. This implies that $l_k(G_1, v_1) \leqslant |q \cap V(G_1)| = 1$ and so $l_k(G_1, v_1) = 1$. Also, $l_k(G_2, v_2) \leqslant |q \cap V(G_2)| < k$.

On the other hand, suppose $l_k(G_1, v_1) = 1$ and $l_k(G_2, v_2) < k$ and $p_k(G_2) = p_k(G_2, v_2)$. Let P_i be an optimal rooted k-path partition of G_i and q_i be the path containing v_i in P_i for $i = 1, 2$. Since $|q_1| = l_k(G_1, v_1) = 1$ and $|q_2| = l_k(G_2, v_2) \leqslant k$, $(P_1 \cup P_2 \cup \{q_1 \cup \{v_1v_2\} \cup q_2\}) - \{q_1, q_2\}$ is a rooted k-path partition of size $p_k(G_1, v_1) + p_k(G_2, v_2) - 1 = p_k(G_1, v_1) + p_k(G_2) - 1$ and so $p_k(G, v_1) = p_k(G_1, v_1) + p_k(G_2) - 1$. □

Theorem 5. *If G is the composition of two graphs G_1 and G_2 with roots v_1 and v_2 respectively, then*

$$l_k(G, v_1) = \begin{cases} l_k(G_2, v_2) + 1 & \text{if } l_k(G_1, v_1) = 1 \text{ and } l_k(G_2, v_2) < k \\ & \text{and } p_k(G_2) = p_k(G_2, v_2), \\ l_k(G_1, v_1) & \text{if } p_k(G_2) \neq p_k(G_2, v_2), \\ \min\{l_k(G_1, v_1), l_k(G_2, v_2) + 1\} & \text{otherwise.} \end{cases}$$

Proof. Suppose P is an optimal rooted k-path partition of G, where the length of the path q containing v_1 is equal to $l_k(G, v_1)$. For the case of $l_k(G_1, v_1) = 1$, $l_k(G_2, v_2) < k$, and $p_k(G_2) = p_k(G_2, v_2)$, from the proof of Theorem 4, it follows that $l_k(G, v_1) = l_k(G_2, v_2) + 1$. For the other cases, $l_k(G, v_1) \leqslant l_k(G_1, v_1)$ from the proof of Lemma 2(b).

Suppose $p_k(G_2) \neq p_k(G_2, v_2)$, i.e., $p_k(G_2) = p_k(G_2, v_2) - 1$ by Lemma 2(a). If $q \cap V(G_2) \neq \emptyset$, then P_{G_i} is a rooted k-path partition of G_i for $i = 1, 2$ and

$$\begin{aligned} |P_{G_1}| + |P_{G_2}| &= p_k(G, v_1) + 1 \\ &= p_k(G_1, v_1) + p_k(G_2) + 1 \\ &= p_k(G_1, v_1) + p_k(G_2, v_2). \end{aligned}$$

It follows that $|P_{G_1}| = p_k(G_1, v_1)$ and $l_k(G_1, v_1) = 1$. Thus, $l_k(G, v_1) = 1 = l_k(G_1, v_1)$. If $q \cap V(G_2) = \emptyset$. Then

$$|P_{G_1}| + |P_{G_2}| = p_k(G, v) = p_k(G_1, v_1) + p_k(G_2).$$

This implies $|P_{G_1}| = p_k(G_1, v_1)$ and $l_k(G_1, v_1) \leqslant l_k(G, v_1)$, i.e., $l_k(G_1, v_1) = l_k(G, v_1)$.

For the last case, $l_k(G, v_1) \leqslant l_k(G_2, v_2) + 1$ since $p_k(G_2) = p_k(G_2, v_2)$. If $l_k(G_1, v_1) = 1$, then $1 \leqslant l_k(G, v_1) \leqslant l_k(G_1, v_1) = 1$. Assume $l_k(G_1, v_1) \neq 1$. Then $|P_{G_1 - v_1}| = p_k(G_1, v_1)$ and $|P_{G_2}| = p_k(G_2, v_2)$. If $q \cap V(G_2) = \emptyset$, then $l_k(G, v_1) = l_k(G_1, v_1)$. Suppose $q \cap V(G_2) \neq \emptyset$. Then $l_k(G, v_1) = l_k(G_2, v_2) + 1$. Therefore, $l_k(G, v_1) = \min\{l_k(G_1, v_1), l_k(G_2, v_2) + 1\}$. □

3. Algorithm

Having proved Theorems 3–5, we are ready to present a linear-time algorithm for finding the k-path partition number of an arbitrary tree.

Algorithm KPPN. *Find the k-path partition number of a tree.*
Input: A tree T and a positive integer k.
Output: The k-path partition number $p_k(T)$.
Method.
Label the vertices of T as $v_1, v_2, \ldots, v_n$ by DFS;

```
for i = 1 to n do
    p(v_i) ← 1; /* for p_k(T_i) */
    p'(v_i) ← 1; /* for p_k(T_i, v_i) */
    l(v_i) ← 1; /* for l_k(T_i, v_i) */
    end do;
for i = 1 to n − 1 do
    let v_j be the ancestor of v_i;
    A ← {t | p'(v_t) = p(v_t), t = i or j};
    if |A| = 2 and l(v_i) + l(v_j) ≤ k
            then p(v_j) ← p(v_j) + p(v_i) − 1
            else p(v_j) ← p(v_j) + p(v_i);
    if l(v_j) = 1 and l(v_i) < k and A = {i}
            then p(v_j) ← p'(v_j) + p(v_i) − 1;
                 l(v_j) ← l(v_i) + 1
            else p(v_j) ← p'(v_j) + p(v_i);
                 if p(v_i) = p'(v_i) then l(v_j) ← min{l(v_j), l(v_i) + 1};
    end do;
output p(v_n).
```

Theorem 6. *Algorithm KPPN computes the k-path partition number of a tree in linear time.*

Proof. It is clear that Algorithm KPPN runs in linear time. The correctness of the algorithm follows from Theorems 3–5. □

Algorithm KPPN can also be modified very simply to produce a minimum path partition of a tree such that each path has weight at most k when vertices and edges have weights.

References

[1] N. Abbas, Graph clustering: complexity, sequential and parallel algorithms, Ph.D. Thesis, Dept. Computing Science, Univ. Alberta, Edmonton, Alberta, Canada, 1994.

[2] S.R. Arikati and C. Pandu Rangan, Linear algorithm for optimal path cover problem on interval graphs, Inform. Process. Lett. 35 (1990) 149–153.

[3] M.A. Bonuccelli and D.P. Bovet, Minimum node disjoint path covering for circular-arc graphs, Inform. Process. Lett. 8 (1979) 159–161.

[4] G.J. Chang and D. Kuo, The L(2,1)-labeling problem on graphs, SIAM J. Discrete Math., to appear.

[5] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs (Academic Press, New York, 1980).

[6] S.E. Goodman and S.T. Hedetniemi, On the Hamiltonian completion problem, Graph Theory and Combinatorics, 1973 (Springer, Berlin, 1974) 262–272.

[7] S.M. Hedetniemi, S.T. Hedetniemi and A.L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1988) 319–349.

[8] W.L. Hsu, An $O(n^2 \log n)$ algorithm for the Hamiltonian cycle problem on circular-arc graphs, SIAM J. Comput. 21 (1992) 1026–1046.

[9] J. Misra and R.E. Tarjan, Optimal chain partitions of trees, Inform. Process. Lett. 4 (1975) 24–26.

[10] Z. Skupien, Path partitions of vertices and hamiltonicity of graphs, in: Proc. 2nd Czechoslovakian Symp. on Graph Theory, Prague (1974).

[11] R. Srikant, Ravi Sundaram, Karan Sher Singh and C. Pandu Rangan, Optimal path cover problem on block graphs and bipartite permutation graphs, Theoret. Comput. Sci. 115 (1993) 351–357.

[12] J.H. Yan, The path partition and related problems, Ph.D. Thesis, Dept. Applied Math., National Chiao Tung Univ., Hsinchu, Taiwan, 1994.

[13] J.H. Yan and G.J. Chang, The path partition problem, Inform. Process. Lett. 52 (1994) 317–322.