

# Path partition for graphs with special blocks<sup>☆</sup>

Jun-Jie Pan<sup>a</sup>, Gerard J. Chang<sup>b, c</sup>

<sup>a</sup>Department of Applied Mathematics, National Chiao Tung University, Hsinchu 30050, Taiwan

<sup>b</sup>Department of Mathematics, National Taiwan University, Taipei 10617, Taiwan

<sup>c</sup>Mathematics Division, National Center for Theoretical Sciences at Taipei, Old Mathematics Building, National Taiwan University, Taipei 10617, Taiwan

Received 8 January 2002; received in revised form 18 August 2003; accepted 18 March 2004

Available online 5 August 2004

## Abstract

The path-partition problem is to find a minimum number of vertex-disjoint paths that cover all vertices of a given graph. This paper studies the path-partition problem from an algorithmic point of view. As the Hamiltonian path problem is NP-complete for many classes of graphs, so is the path-partition problem. The main result of this paper is to present a linear-time algorithm for the path-partition problem in graphs whose blocks are complete graphs, cycles or complete bipartite graphs.

© 2004 Elsevier B.V. All rights reserved.

**Keywords:** Path partition; Block; Complete graph; Cycle; Complete bipartite graph; Algorithm

## 1. Introduction

A *path partition* of a graph is a collection of vertex-disjoint paths that cover all vertices of the graph. The *path-partition problem* is to find the *path-partition number*  $p(G)$  of a graph  $G$ , which is the minimum cardinality of a path partition of  $G$ . Notice that  $G$  has a Hamiltonian path if and only if  $p(G) = 1$ . Since the Hamiltonian path problem is NP-complete for planar graphs [9], bipartite graphs [10], chordal graphs [10], chordal bipartite graphs [14] and strongly chordal graphs [14], so is the path-partition problem. On the other hand, the path-partition problem is polynomially solvable for trees [11,16], interval graphs [1,2,7], circular-arc graphs [2,7], cographs [5,6,13], cocomparability graphs [8], block graphs [17–19] and bipartite distance-hereditary graphs [21]. For some references of related problems, see [3,4,12,15,20].

The purpose of this paper is to give a linear-time algorithm for the path-partition problem for graphs whose blocks are complete graphs, cycles or complete bipartite graphs. For technical reasons, we consider the following generalized problem, which is a *labeling approach* for the problem.

Suppose every vertex  $v$  in the graph  $G$  is associated with an integer  $f(v) \in \{0, 1, 2, 3\}$ . An  *$f$ -path partition* is a collection  $\mathcal{P}$  of vertex-disjoint paths such that the following conditions hold:

- (P1) Any vertex  $v$  with  $f(v) \neq 3$  is in some path in  $\mathcal{P}$ .
- (P2) If  $f(v) = 0$ , then  $v$  itself is a path in  $\mathcal{P}$ .
- (P3) If  $f(v) = 1$ , then  $v$  is an end vertex of some path in  $\mathcal{P}$ .

*E-mail address:* [gjchang@math.ntu.edu.tw](mailto:gjchang@math.ntu.edu.tw) (G.J. Chang).

<sup>☆</sup> Supported in part by the National Science Council under grant NSC90-2115-M002-024.

The  $f$ -path-partition problem is to determine the  $f$ -path-partition number  $p_f(G)$  which is the minimum cardinality of an  $f$ -path partition of  $G$ . It is clear that  $p(G) = p_f(G)$  when  $f(v) = 2$  for all vertices  $v$  in  $G$ .

In the rest of this section, we review some terminology in graphs. A *cut-vertex* is a vertex whose removal results in a graph having more components than the original graph. A *block* is a maximal connected subgraph without a cut-vertex. Notice that the intersection of two distinct blocks contains at most one vertex; and a vertex is a cut-vertex if and only if it is the intersection of two or more blocks. Consequently, a graph with one or more cut-vertices has at least two blocks. An *end block* is a block with exactly one cut-vertex.

## 2. Path partition in graphs

The labeling approach used in this paper starts from the end blocks. Suppose  $B$  is an end block whose only cut-vertex is  $x$ . Let  $A$  be the graph  $G - (V(B) - \{x\})$ . Notice that we can view  $G$  as the “composition” of  $A$  and  $B$ , i.e.,  $G$  is the union of  $A$  and  $B$  which meet at a common vertex  $x$ . The idea is to get the path-partition number of  $G$  from those of  $A$  and  $B$ .

In the lemmas and theorems of this paper, we use the following notation. Suppose  $x$  is a specified vertex of a graph  $H$  in which  $f$  is a vertex labeling. For  $i = 0, 1, 2, 3$ , we define the function  $f_i : V(H) \rightarrow \{0, 1, 2, 3\}$  by  $f_i(y) = f(y)$  for all vertices  $y$  except  $f_i(x) = i$ .

**Lemma 1.** *Suppose  $x$  is a specified vertex in a graph  $H$ . Then the following statements hold.*

- (1)  $p_{f_3}(H) \leq p_{f_2}(H) \leq p_{f_1}(H) \leq p_{f_0}(H)$ .
- (2)  $p_{f_1}(H) \leq p_{f_0}(H) \leq p_{f_1}(H) + 1$ .
- (3)  $p_{f_2}(H) \leq p_{f_1}(H) \leq p_{f_2}(H) + 1$ .
- (4)  $p_{f_3}(H) = \min\{p_{f_2}(H), p_f(H - x)\} \leq p_f(H - x) = p_{f_0}(H) - 1$ .
- (5)  $p_f(H) \geq p_{f_1}(H) - 1$ .

**Proof.** (1) The inequalities follow from that an  $f_i$ -path partition is an  $f_j$ -path partition whenever  $i < j$ .

(2) The second inequality follows from that replacing the path  $Px$  in an  $f_1$ -path partition by two paths  $P$  and  $x$  results an  $f_0$ -path partition of  $H$ .

(3) The second inequality follows from that replacing the path  $PxQ$  in an  $f_2$ -path partition by two paths  $Px$  and  $Q$  results an  $f_1$ -path partition of  $H$ .

(4) The first equality follows from that one is an  $f_3$ -path partition of  $H$  if and only if it is either an  $f_2$ -path partition of  $H$  or an  $f$ -path partition of  $H - x$ . The second equality follows from that  $\mathcal{P}$  is an  $f_0$ -path partition of  $H$  if and only if it is the union of  $\{x\}$  and an  $f$ -path partition of  $H - x$ .

(5) According to (1), (3) and (4), we have

$$p_f(H) \geq p_{f_3}(H) = \min\{p_{f_2}(H), p_f(H - x)\} \geq \min\{p_{f_1}(H) - 1, p_{f_0}(H) - 1\} = p_{f_1}(H) - 1. \quad \square$$

**Lemma 2.** (1)  $p_f(G) \leq \min\{p_f(A) + p_{f_0}(B) - 1, p_{f_0}(A) + p_f(B) - 1\}$ .

(2)  $p_{f_2}(G) \leq p_{f_1}(A) + p_{f_1}(B) - 1$ .

**Proof.** (1) Suppose  $\mathcal{P}$  is an optimal  $f$ -path partition of  $A$ , and  $\mathcal{Q}$  an  $f_0$ -path partition of  $B$ . Then  $x \in \mathcal{Q}$  and so  $(\mathcal{P} \cup \mathcal{Q}) - \{x\}$  is an  $f$ -path partition of  $G$ . This gives  $p_f(G) \leq p_f(A) + p_{f_0}(B) - 1$ . Similarly,  $p_f(G) \leq p_{f_0}(A) + p_f(B) - 1$ .

(2) The inequality follows from that if  $\mathcal{P}$  (respectively,  $\mathcal{Q}$ ) is an optimal  $f_1$ -path partition of  $A$  (respectively,  $B$ ) in which  $Px \in \mathcal{P}$  (respectively,  $xQ \in \mathcal{Q}$ ) contains  $x$ , then  $(\mathcal{P} \cup \mathcal{Q} \cup \{PxQ\}) - \{Px, xQ\}$  is an  $f_2$ -path partition of  $G$ .  $\square$

We now have the following theorem which is key for the inductive step of our algorithm.

**Theorem 3.** *Suppose  $\alpha = p_{f_0}(B) - p_{f_1}(B)$  and  $\beta = p_{f_1}(B) - p_{f_2}(B)$ . (Notice that  $\alpha, \beta \in \{0, 1\}$ .) Then the following statements hold:*

- (1) If  $f(x) = 0$ , then  $p_f(G) = p_f(A) + p_f(B) - 1$ .
- (2) If  $f(x) = 1$ , then  $p_f(G) = p_{f_1-\alpha}(A) + p_{f_2}(B) - 1$ .
- (3) If  $f(x) \geq 2$  and  $\alpha = \beta = 0$ , then  $p_f(G) = p_f(A) + p_{f_0}(B) - 1$ .
- (4) If  $f(x) \geq 2$  and  $\alpha = 0$  and  $\beta = 1$ , then  $p_f(G) = p_{f_3}(A) + p_f(B)$ .
- (5) If  $f(x) \geq 2$  and  $\alpha = 1$ , then  $p_f(G) = p_{f_1-\beta}(A) + p_{f_1+\beta}(B) - 1$ .

**Proof.** Suppose  $\mathcal{P}$  is an optimal  $f$ -path partition of  $G$ . Let  $P^*$  be the path in  $\mathcal{P}$  that contains  $x$ . (It is possible that there is no such path when  $f(x) = 3$ .) There are three possibilities for  $P^*$ : (a)  $P^*$  does not exist or  $P^* \subseteq A$ ; (b)  $P^* \subseteq B$ ; (c)  $x$  is an internal vertex of  $P^*$ , say  $P^* = P'xP''$ , with  $P'x \subseteq A$  and  $xP'' \subseteq B$ . (The latter is possible only when  $f(x) \geq 2$ .)

For the case when (a) holds,  $\{P \in \mathcal{P} : P \subseteq A\}$  is an  $f$ -path partition of  $A$  and  $\{P \in \mathcal{P} : P \subseteq B\} \cup \{x\}$  is an  $f_0$ -path partition of  $B$ . We then have the inequality in (a'). Similarly, we have (b') and (c') corresponding to (b) and (c).

$$(a') \quad p_f(G) \geq p_f(A) + p_{f_0}(B) - 1.$$

$$(b') \quad p_f(G) \geq p_{f_0}(A) + p_f(B) - 1. \text{ (We may replace } p_f(B) \text{ by } p_{f_2}(B) \text{ when } f(x) \geq 2.)$$

$$(c') \quad p_f(G) \geq p_{f_1}(A) + p_{f_1}(B) - 1. \text{ (This is possible only when } f(x) \geq 2.)$$

We are now ready to prove the theorem.

(1) Since  $f(x) = 0$ , we have  $f = f_0$ . According to Lemma 2(1),  $p_f(G) \leq p_f(A) + p_f(B) - 1$ . On the other hand, (a') and (b') give  $p_f(G) \geq p_f(A) + p_f(B) - 1$ .

(2) Since  $f(x) = 1$ , we have  $f = f_1$ . Lemma 2(1), together with (a') and (b'), gives  $p_f(G) = \min\{p_{f_1}(A) + p_{f_0}(B) - 1, p_{f_0}(A) + p_{f_1}(B) - 1\}$ . If  $\alpha = 0$ , then

$$p_{f_0}(A) + p_{f_1}(B) - 1 \geq p_{f_1}(A) + (p_{f_0}(B) - \alpha) - 1 = p_{f_1}(A) + p_{f_0}(B) - 1;$$

and if  $\alpha = 1$ , then

$$p_{f_1}(A) + p_{f_0}(B) - 1 \geq (p_{f_0}(A) - 1) + (p_{f_1}(B) + \alpha) - 1 = p_{f_0}(A) + p_{f_1}(B) - 1.$$

Hence  $p_f(G) = p_{f_1-\alpha}(A) + p_{f_\alpha}(B) - 1$ .

(3) According to Lemma 2(1),  $p_f(G) \leq p_f(A) + p_{f_0}(B) - 1$ . On the other hand, as  $p_{f_0}(A) \geq p_{f_1}(A) \geq p_f(A)$  and  $p_{f_0}(B) = p_{f_1}(B) = p_{f_2}(B)$ , (a')–(c') give  $p_f(G) \geq p_f(A) + p_{f_0}(B) - 1$ .

(4) According to Lemma 1(4) and  $\alpha = 0$  and  $\beta = 1$ , we have

$$p_f(B - x) = p_{f_0}(B) - 1 = p_{f_1}(B) - 1 = p_{f_2}(B).$$

This, together with Lemma 1(4), gives that the above value is also equal to  $p_{f_3}(B)$  and so  $p_f(B)$ . Then, an optimal  $f_3$ -path partition  $\mathcal{P}$  of  $A$ , together with an optimal  $p_f$ -path partition of  $B - x$  (respectively,  $B$ ) when  $x$  is (respectively, is not) in a path of  $\mathcal{P}$ , forms an  $f_2$ -path partition of  $G$ . Thus,  $p_f(G) \leq p_{f_2}(G) \leq p_{f_3}(A) + p_f(B)$ .

On the other hand, since  $p_{f_1}(A) \geq p_f(A) \geq p_{f_3}(A)$  and  $p_{f_0}(B) - 1 = p_{f_1}(B) - 1 = p_f(B)$ , (a') or (c') implies  $p_f(G) \geq p_{f_3}(A) + p_f(B)$ . Also, as  $p_{f_0}(A) - 1 \geq p_{f_3}(A)$  by Lemma 1(4), (b') implies  $p_f(G) \geq p_{f_3}(A) + p_f(B)$ .

(5) According to Lemma 1(1) and Lemma 2, we have

$$p_f(G) \leq p_{f_2}(G) \leq \min\{p_{f_0}(A) + p_{f_2}(B) - 1, p_{f_1}(A) + p_{f_1}(B) - 1\}.$$

On the other hand, if (a') holds, then by Lemma 1(5) and that  $p_{f_0}(B) = p_{f_1}(B) + 1$ ,

$$p_f(G) \geq p_f(A) + p_{f_0}(B) - 1 \geq (p_{f_1}(A) - 1) + (p_{f_1}(B) + 1) - 1 = p_{f_1}(A) + p_{f_1}(B) - 1.$$

This, together with (b') and (c'), gives

$$p_f(G) = \min\{p_{f_0}(A) + p_{f_2}(B) - 1, p_{f_1}(A) + p_{f_1}(B) - 1\}.$$

If  $\beta = 0$ , then

$$p_{f_0}(A) + p_{f_2}(B) - 1 \geq p_{f_1}(A) + (p_{f_1}(B) - \beta) - 1 = p_{f_1}(A) + p_{f_1}(B) - 1;$$

and if  $\beta = 1$ , then

$$p_{f_1}(A) + p_{f_1}(B) - 1 \geq (p_{f_0}(A) - 1) + (p_{f_2}(B) + \beta) - 1 = p_{f_0}(A) + p_{f_2}(B) - 1.$$

Hence  $p_f(G) = p_{f_1-\beta}(A) + p_{f_1+\beta}(B) - 1$ .  $\square$

### 3. Special blocks

Notice that the inductive theorem (Theorem 3) can be applied to solve the path-partition problem on graphs for which the problem can be solved on its blocks. In this paper, we mainly consider the case when the blocks are complete graphs, cycles or complete bipartite graphs.

Now, we assume that  $B$  is a graph in which each vertex  $v$  has a label  $f(v) \in \{0, 1, 2, 3\}$ . Recall that  $f^{-1}(i)$  is the set of preimages of  $i$ , i.e.

$$f^{-1}(i) = \{v \in V(B) : f(v) = i\}.$$

According to Lemma 1(4), we have  $p_f(B) = p_f(B - f^{-1}(0)) + |f^{-1}(0)|$ . Therefore, we may assume without loss of generality that  $f^{-1}(0) = \emptyset$  throughout this section.

We first consider the case when  $B$  is a complete graph. The proof of the following lemma is straightforward and hence omitted.

**Lemma 4.** *Suppose  $B$  is a complete graph. If  $f^{-1}(1) \neq \emptyset$  or  $f^{-1}(2) = \emptyset$ , then  $p_f(B) = \lceil |f^{-1}(1)|/2 \rceil$  else  $p_f(B) = 1$ .*

Next, consider the case when  $B$  is a path. This is useful as a subroutine for handling cycles. The proof of the following lemma is also omitted.

**Lemma 5.** *Suppose  $B$  is a path.*

- (1) *If  $x$  is an end vertex of  $B$  with  $f(x) = 3$ , then  $p_f(B) = p_f(B - x)$ .*
- (2) *If  $x$  is an end vertex of  $B$  with  $f(x) = 2$ , then  $p_f(B) = p_{f_1}(B)$ .*
- (3) *If  $B$  has an end vertex  $x$  and another vertex  $y$  with  $f(x) = f(y) = 1$  such that no vertex between  $x$  and  $y$  has a label 1, then  $p_f(B) = p_f(B') + 1$  where  $B'$  is the path obtained from  $B$  by deleting  $x$ ,  $y$  and all vertices between them.*

We then consider the case when  $B$  is a cycle. The proof of the following lemma is also omitted.

**Lemma 6.** *Suppose  $B$  is a cycle.*

- (1) *If  $f^{-1}(2) = \emptyset$ , then  $p_f(B) = \lceil |f^{-1}(1)|/2 \rceil$ .*
- (2) *If  $P$  is a path from  $x$  to  $y$  in  $B$  such that  $f^{-1}(1) \cap P = \{x, y\}$  and  $f^{-1}(2) \cap P \neq \emptyset$ , then  $p_f(B) = p_f(B - P) + 1$ .*

Finally, we consider the case when  $B$  is a complete bipartite graph with  $C \cup D$  as a bipartition of the vertex set. For  $i \in \{0, 1, 2, 3\}$ , let

$$C_i = \{u \in C : f(u) = i\} \quad \text{with } c_i = |C_i|;$$

$$D_i = \{v \in D : f(v) = i\} \quad \text{with } d_i = |D_i|.$$

We have the following lemmas.

**Lemma 7.** *If  $c_1 = d_1 = 0$  and  $c_2 \geq d_2$  and  $x \in C_2$ , then  $p_f(B) = p_{f'}(B)$  where  $f'$  is the same as  $f$  except  $f'(x) = 1$ .*

**Proof.**  $p_f(B) \leq p_{f'}(B)$  follows from the fact that any  $f'$ -path partition of  $B$  is an  $f$ -partition.

Suppose  $\mathcal{P}$  is an optimal  $f$ -path partition of  $B$ . We may assume that  $\mathcal{P}$  is chosen so that the paths in  $\mathcal{P}$  cover as few vertices as possible. For the case when  $\mathcal{P}$  has a path  $P_y$  with  $y \in C$ , we may interchange  $y$  and  $x$  to assume that  $P_x \in \mathcal{P}$ . In this case,  $\mathcal{P}$  is an  $f'$ -path partition of  $B$  and so  $p_{f'}(B) \leq p_f(B)$ . So, now assume that all end vertices of paths in  $\mathcal{P}$  are in  $D$ . Then, these end vertices are all in  $D_2$  for otherwise we may delete those end vertices in  $D_3$  to get a new  $\mathcal{P}$  which covers fewer vertices. We may further assume that paths in  $\mathcal{P}$  cover no vertices in  $D_3$ , for otherwise we may interchange such a vertex with an end vertex of a path in  $\mathcal{P}$  and then delete it from the path. Thus each path of  $\mathcal{P}$  uses vertices in  $C_2 \cup C_3 \cup D_2$ , and has end vertices in  $D_2$ . These imply that  $d_2 > c_2$ , contradicting that  $c_2 \geq d_2$ .  $\square$

By symmetry, we may prove a similar theorem for the case when  $d_1 = c_1 = 0$  and  $d_2 \geq c_2$  and  $d_2 \geq 1$ .

**Lemma 8.** Suppose  $x \in C_1$ . Also, either  $d_2 \geq 1$  with  $y \in D_2$ , or else  $c_1 > d_1$  and  $d_2 = 0 < d_3$  with  $y \in D_3$ . Then  $p_f(B) = p_{f'}(B - x)$ , where  $f'$  is the same as  $f$  except  $f'(y) = 1$ .

**Proof.** Suppose  $P_y$  is in an optimal  $f'$ -path partition  $\mathcal{P}$  of  $B - x$ . Then  $(\mathcal{P} - \{P_y\}) \cup \{P_yx\}$  is an  $f$ -path partition of  $B$  and so  $p_f(B) \leq p_{f'}(B - x)$ .

On the other hand, suppose  $P_x$  is in an optimal  $f$ -path partition  $\mathcal{P}$  of  $B$ . For the case when  $y$  is not covered by any path in  $\mathcal{P}$ , we have  $y \in D_3$  and so  $c_1 > d_1$  and  $d_2 = 0$ . Consequently, there is some  $Qz \in \mathcal{P}$  with  $z \in C_2 \cup C_3$  or  $z \in D_3$ . For the former case, we replace  $Qz$  by  $Qzy$  in  $\mathcal{P}$ ; for the later, we replace  $Qz$  by  $Qy$ . So, in any case we may assume that  $y$  is covered by some path  $RyS$  in  $\mathcal{P}$ . If  $RyS = P_x$ , then again we may interchange  $y$  with the last vertex of  $P$  to assume that  $RyS = Tyx$  in  $\mathcal{P}$  for some  $T$ . If  $RyS \neq P_x$ , then we may replace the two paths  $RyS$  and  $P_x$  by  $Ryx$  and  $PS$ . So, in any case, we may assume that  $\mathcal{P}$  has a path  $Uyx$ . Then,  $(\mathcal{P} - \{Uyx\}) \cup \{Uy\}$  is an  $f'$ -path partition of  $B - x$ . Thus  $p_{f'}(B - x) \leq p_f(B)$ .  $\square$

By symmetry, we may prove a similar theorem for the case when  $x \in D_1$ ; and either  $c_2 \geq 1$  with  $y \in C_2$ , or else  $d_1 > c_1$  and  $c_2 = 0 < c_3$  with  $y \in C_3$ .

#### 4. Algorithm

We are ready to give a linear-time algorithm for the path-partition problem in graphs whose blocks are complete graphs, cycles or complete bipartite graphs. Notice that we may consider only connected graphs. We present five procedures. The first four are subroutines which calculate  $f$ -path-partition numbers of complete graphs, paths, cycles and complete bipartite graphs, respectively, by using Lemmas 4–8. The last one is the main routine for the problem.

First, Lemmas 1(4) and 4 lead to the following subroutine for complete graphs.

**Algorithm PCG.** Find the  $f$ -path partition number  $p_f(B)$  of a complete graph  $B$ .

**Input.** A complete graph  $B$  and a vertex labeling  $f$ .

**Output.**  $p_f(B)$ .

**Method.**

```

if  $(f^{-1}(1) \neq \emptyset$  or  $f^{-1}(2) = \emptyset)$ 
then  $p_f(B) = |f^{-1}(0)| + \lceil |f^{-1}(1)|/2 \rceil$ ;
else  $p_f(B) = |f^{-1}(0)| + 1$ ;
return  $p_f(B)$ .

```

Lemma 5 leads to the following subroutine for paths, which is useful for the cycle subroutine.

**Algorithm PP.** Find the  $f$ -path partition number  $p_f(B)$  of the path  $B$ .

**Input.** A path  $B$  and a vertex labeling  $f$  with  $f^{-1}(0) = \emptyset$ .

**Output.**  $p_f(B)$ .

**Method.**

```

 $p_f(B) \leftarrow 0$ ;
 $B' \leftarrow B$ ;
while  $(B' \neq \emptyset)$  do
  choose an end vertex  $x$  of  $B'$ ;
  if  $(f(x) = 3)$  then  $B' \leftarrow B' - x$  else
    choose a vertex  $y$  nearest to  $x$  with  $f(y) = 1$ 
    (let  $y$  be the other end vertex if there is no such vertex);
     $p_f(B) \leftarrow p_f(B) + 1$ ;
     $B' \leftarrow B' -$  all vertices between (and including)  $x$  and  $y$ ;
  end else;
end while;
return  $p_f(B)$ .

```

Lemmas 1(4) and 6 lead to the following subroutine for cycles.

**Algorithm PC.** Find the  $f$ -path partition number  $p_f(B)$  of a cycle  $B$ .

**Input.** A cycle  $B$  and a vertex labeling  $f$ .

**Output.**  $p_f(B)$ .

**Method.**

```

if  $(f^{-1}(0) = \emptyset$  and  $f^{-1}(2) = \emptyset)$ 
then  $p_f(B) \leftarrow \lceil f^{-1}(1)/2 \rceil$ ;
else if  $(f^{-1}(0) = \emptyset$  and  $f^{-1}(2) \neq \emptyset$  and  $|f^{-1}(1)| \leq 1)$  then
     $p_f(B) \leftarrow 1$ ;
else if  $(f^{-1}(0) = \emptyset$  and  $f^{-1}(2) \neq \emptyset$  and  $|f^{-1}(1)| \geq 2)$  then
    choose a path  $P$  from  $x$  to  $y$  such that
         $f^{-1}(1) \cap P = \{x, y\}$  and  $f^{-1}(2) \cap P \neq \emptyset$ ;
     $p_f(B) \leftarrow p_f(B - P) + 1$  by calling PP $(B - P)$ ;
else // now  $f^{-1}(0) \neq \emptyset$  //
    let  $B - f^{-1}(0)$  be the disjoint union of paths  $P_1, P_2, \dots, P_k$ ;
     $p_f(B) \leftarrow |f^{-1}(0)|$ ;
    for  $i = 1$  to  $k$  do  $p_f(B) \leftarrow p_f(B) + p_f(P_i)$  by calling PP $(P_i)$ ;
end else;
return  $p_f(B)$ .

```

Lemmas 1(4), 7 and 8 lead to the following subroutine for complete bipartite graphs. In the subroutine, we inductively reduce the size of  $C \cup D$ . Besides the reduction of  $C_0$  and  $D_0$  in the second line, we consider 9 cases. The first case is for  $C = \emptyset$  or  $D = \emptyset$ . The next 5 cases are for  $c_1 \geq 1$  or  $d_1 \geq 1$ . In particular, the case of  $c_1 \geq 1$  is covered by cases 2 and 3, except when  $d_2 = 0$  and  $(c_1 \leq d_1$  or  $d_3 = 0)$ . The case of  $d_1 \geq 1$  is covered by cases 4 and 5, except when  $c_2 = 0$  and  $(d_1 \leq c_1$  or  $c_3 = 0)$ . The exceptions are then covered by case 6. Finally, the last 3 cases are for  $c_1 = d_1 = 0$ .

**Algorithm PCB.** Find the  $f$ -path partition number  $p_f(B)$  of a complete bipartite graph  $B$ .

**Input:** A complete bipartite graph  $B$  with a bipartition  $C \cup D$  of vertices and a vertex labeling  $f$ .

**Output:**  $p_f(B)$ .

**Method.**

```

 $c_i \leftarrow |f^{-1}(i) \cap C|$  and  $d_i \leftarrow |f^{-1}(i) \cap D|$  for  $0 \leq i \leq 3$ ;
 $p_f(B) \leftarrow c_0 + d_0$ ;
while (true) do
    if  $(c_1 = c_2 = c_3 = 0$  or  $d_1 = d_2 = d_3 = 0)$  then
         $p_f(B) \leftarrow p_f(B) + c_1 + c_2 + d_1 + d_2$ ; return  $p_f(B)$ ;
    else if  $(c_1 \geq 1$  and  $d_2 \geq 1)$  then // use Lemma 8 //
         $c_1 \leftarrow c_1 - 1$ ;  $d_2 \leftarrow d_2 - 1$ ;  $d_1 \leftarrow d_1 + 1$ ;
    else if  $(c_1 \geq 1$  and  $c_1 > d_1$  and  $d_2 = 0 < d_3)$  then // use Lemma 8 //
         $c_1 \leftarrow c_1 - 1$ ;  $d_3 \leftarrow d_3 - 1$ ;  $d_1 \leftarrow d_1 + 1$ ;
    else if  $(d_1 \geq 1$  and  $c_2 \geq 1)$  then // use the remark after Lemma 8 //
         $d_1 \leftarrow d_1 - 1$ ;  $c_2 \leftarrow c_2 - 1$ ;  $c_1 \leftarrow c_1 + 1$ ;
    else if  $(d_1 \geq 1$  and  $d_1 > c_1$  and  $c_2 = 0 < c_3)$  then // use the remark after Lemma 8 //
         $d_1 \leftarrow d_1 - 1$ ;  $c_3 \leftarrow c_3 - 1$ ;  $c_1 \leftarrow c_1 + 1$ ;
    else if  $(c_2 = d_2 = 0$  and  $(c_1 = d_1 \geq 1$  or  $c_1 > d_1 \geq 1$  with  $d_3 = 0$  or  $d_1 > c_1 \geq 1$  with  $c_3 = 0))$  then
         $p_f(B) \leftarrow p_f(B) + \max\{c_1, d_1\}$ ; return  $p_f(B)$ ;
    else // by now  $c_1 = d_1 = 0$  // if  $(c_2 = d_2 = 0)$  then
        return  $p_f(B)$ ;
    else if  $(c_2 \geq d_2)$  then // use Lemma 7 //
         $c_1 \leftarrow 1$ ;  $c_2 \leftarrow c_2 - 1$ ;
    else if  $(c_2 < d_2)$  then // use the remark after Lemma 7 //
         $d_1 \leftarrow 1$ ;  $d_2 \leftarrow d_2 - 1$ ;
end while.

```

Finally, Theorem 3 together with the subroutines above leads to the following main algorithm.

**Algorithm PG.** Find the path-partition number  $p_f(G)$  of the connected graph  $G$  whose blocks are cycles, complete graphs or complete bipartite graphs.

**Input:** A graph  $G$  and a vertex labeling  $f$ .

**Output:**  $p_f(G)$ .

**Method.**

```

 $p_f(G) \leftarrow 0;$ 
 $G' \leftarrow G;$ 
while ( $G' \neq \emptyset$ ) do
  choose a block  $B$  of  $G'$  with only one cut-vertex  $x$  or with no cut-vertex;
  if ( $B$  is a complete graph) then
    find  $p_{f_i}(B)$  by calling PCG( $B, f_i$ ) for  $0 \leq i \leq 3$ ;
  if ( $B$  is a cycle) then
    find  $p_{f_i}(B)$  by calling PC( $B, f_i$ ) for  $0 \leq i \leq 3$ ;
  if ( $B$  is a complete bipartite graph) then
    find  $p_{f_i}(B)$  by calling PCB( $B, f_i$ ) for  $0 \leq i \leq 3$ ;
   $\alpha := p_{f_0}(B) - p_{f_1}(B);$ 
   $\beta := p_{f_1}(B) - p_{f_2}(B);$ 
  if ( $f(x) = 0$ ) then
     $p_f(G) \leftarrow p_f(G) + p_f(B) - 1;$ 
  else if ( $f(x) = 1$ ) then
     $p_f(G) \leftarrow p_f(G) + p_{f_x}(B) - 1; f(x) \leftarrow 1 - \alpha;$ 
  else // by now  $f(x) = 2$  or  $3$  //
    case 1:  $\alpha = \beta = 0$ 
       $p_f(G) \leftarrow p_f(G) + p_{f_0}(B) - 1;$ 
    case 2:  $\alpha = 0$  and  $\beta = 1$ 
       $p_f(G) \leftarrow p_f(G) + p_f(B); f(x) \leftarrow 3;$ 
    case 3:  $\alpha = 1$ 
       $p_f(G) \leftarrow p_f(G) + p_{f_{1+\beta}}(B) - 1; f(x) \leftarrow 1 - \beta;$ 
   $G' := G' - (B - \{x\});$ 
end while;
output  $p_f(G)$ .

```

**Theorem 9.** Algorithm PG computes the  $f$ -path partition number of a connected graph whose blocks are cycles, complete graphs or complete bipartite graphs in linear time.

**Proof.** The correctness of the algorithm follows from Lemma 1(4) and Lemmas 4 to 8. The algorithm takes only linear time since depth-first search can be used to find end blocks and each subroutine requires only  $O(|B|)$  operations.  $\square$

## Acknowledgements

The authors thank the referees for many constructive suggestions.

## References

- [1] S.R. Arikati, C. Pandu Rangan, Linear algorithm for optimal path cover problem on interval graphs, Inform. Process. Lett. 35 (1990) 149–153.
- [2] H.J. Bonuccelli, D.P. Bovet, Minimum node disjoint path covering for circular-arc graphs, Inform. Process. Lett. 8 (1979) 159–161.
- [3] G.J. Chang, Algorithmic aspects of linear  $k$ -arboricity, Taiwanese J. Math. 3 (1999) 73–81.
- [4] G.J. Chang, Corrigendum for ‘The path-partition problem in block graphs’, Inform. Process. Lett. 83 (2002) 293.
- [5] G.J. Chang, D. Kuo, The  $L(2, 1)$ -labeling problem on graphs, SIAM J. Discrete Math. 9 (1996) 309–316.
- [6] D.G. Corneil, H. Lerchs, L. Stewarts, Complement reducible graphs, Discrete Appl. Math. 3 (1981) 163–174.
- [7] P. Damaschke, Paths in interval graphs and circular arc graphs, Discrete Math. 112 (1993) 49–64.
- [8] P. Damaschke, J.S. Deogun, D. Kratsch, G. Steiner, Finding Hamiltonian paths in cocomparability graphs using the bump number algorithm, Order 8 (1992) 383–391.

- [9] M.R. Garey, D.S. Johnson, R.E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, *SIAM J. Comput.* 5 (1976) 704–714.
- [10] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
- [11] H.A. Jung, On a class of posets and the corresponding comparability graphs, *J. Combin. Theory Ser. B* 35 (1978) 125–133.
- [12] Y.D. Liang, G.K. Mancher, C. Rhee, T. Mankus, in: A linear algorithm for finding Hamiltonian circuits in circular-arc graphs, 32nd ACM Southeastern Conference, 1994, pp. 101–118.
- [13] R. Lin, S. Olariu, G. Pruesse, An optimal path cover algorithm for cographs, *Comput. Math. Appl.* 30 (1995) 75–83.
- [14] H. Müller, Hamiltonian circuits in chordal bipartite graphs, *Discrete Math.* 156 (1996) 291–298.
- [15] J.-J. Pan, G.J. Chang, Isometric path numbers of block graphs, submitted.
- [16] Z. Skupien, Path partitions of vertices and Hamiltonicity of graphs, in: *Proceedings of the Second Czechoslovakian Symposium on Graph Theory*, Prague, 1974.
- [17] R. Strikant, Ravi Sundaram, Karan Sher Singh, C. Pandu Rangan, Optimal path cover problem on block graphs and bipartite permutation graphs, *Theoret. Comput. Sci.* 115 (1993) 351–357.
- [18] P.-K. Wong, Optimal path cover problem on block graphs, *Theoret. Comput. Sci.* 225 (1999) 163–169.
- [19] J.-H. Yan, G.J. Chang, The path-partition problem in block graphs, *Inform. Process. Lett.* 52 (1994) 317–322.
- [20] J.-H. Yan, G.J. Chang, S.M. Hedetniemi, S.T. Hedetniemi,  $k$ -Path partitions in trees, *Discrete Appl. Math.* 78 (1997) 227–233.
- [21] H.-G. Yeh, G.J. Chang, The path-partition problem in bipartite distance-hereditary graphs, *Taiwanese J. Math.* 2 (1998) 353–360.