Theoretical Computer Science 115 (1993) 351–357
Elsevier

Note

Optimal path cover problem on block graphs and bipartite permutation graphs

R. Srikant, Ravi Sundaram, Karan Sher Singh and
C. Pandu Rangan

Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600036, India

Communicated by M. Nivat
Received July 1990
Revised August 1992

Abstract

Srikant, R. et al., Optimal path cover problem on block graphs and bipartite permutation graphs, Theoretical Computer Science 115 (1993) 351–357.

The optimal path cover problem is to find a minimum number of vertex disjoint paths which together cover all the vertices of the graph. In this paper, we present linear-time algorithms for the optimal path cover problem for the class of block graphs and bipartite permutation graphs.

1. Introduction

The optimal path cover problem is to find a minimum number of vertex disjoint paths which together cover all the vertices of the graph. Finding an optimal path cover for an arbitrary graph is known to be NP-complete [3]. However, polynomial-time algorithms exist for trees [4], cacti [4] and for interval graphs [9]. The solution presented in [2] for circular-arc graphs is known to be wrong. In this paper, we

Correspondence to: C. Pandu Rangan, Department of Computer Science and Engineering, Indian Institute of Technology, Madras 600036, India. Email: rangan@iitm.ernet.in.

present linear-time algorithms for finding an optimal path cover on bipartite permutation graphs and block graphs. The path cover problem finds applications in establishing ring protocols, code optimization, and mapping parallel programs to parallel architectures [4].

2. Bipartite permutation graphs

In this section, we assume that the given graph $G=(S, T, E)$ is a bipartite permutation graph [6], where S, T are the partite sets of the graph G.

Definition 2.1. A *strong ordering* of the vertices of a bipartite graph $G=(S, T, E)$ consists of an ordering of S and an ordering of T such that for all (s, t), (s', t') in E, $s<s'$ and $t>t'$ imply that $(s, t'), (s', t) \in E$.

Lemma 2.2 (Spinrad et al. [7]). *Let $G=(S, T, E)$ be a bipartite graph. Then the following statements are equivalent:*

(1) *G is a bipartite permutation graph.*

(2) *There is a strong ordering of vertices of G.*

2.1. *Optimal path cover in bipartite permutation graphs*

Definition 2.3. Let G be a bipartite permutation graph. A path cover $(P_1, P_2, \ldots, P_k)$ on G is said to be *contiguous* if it satisfies the following two conditions:

(1) If s is the only vertex in P_i and if $s'<s<s''$, then s' and s'' belong to different paths.

(2) If st is an edge in P_i and $s't'$ is an edge in P_j, where $i \neq j$ and $s<s'$, then $t<t'$.

Lemma 2.4. *Let G be a bipartite permutation graph. Then there exists a contiguous path cover for G which is optimal.*

Proof. We will convert an arbitrary optimal path cover P into a contiguous optimal path cover as follows: We consider the two conditions in Definition 2.3 separately.

(1) Let s', s, s'' be the vertices not satisfying condition 1. Without loss of generality, assume that s', s'' are closest to s from the right and left in the ordering of S in some path P_j of P. Let s'–t–s'' be the subpath of P_j. From Definition 2.1, t is adjacent to s. Connect t with s and remove the connection between t and s'' in the path cover. By repeating this procedure, we get an optimal path cover satisfying condition 1.

(2) Let st and $s't'$, respectively, be the edges in P_i and P_j of P, not satisfying condition 2. From Definition 2.1, we know that st' and $s't$ are in E. Remove the edges st and $s't'$ from the path cover and replace them with st' and $s't$. We get two new paths P_i' and P_j' which cover the same vertices as P_i and P_j. By repeating this for all pairs of

edges which do not satisfy condition 2, we get an optimal path cover which satisfies condition 2.

Hence, we can convert an arbitrary optimal path cover into a contiguous optimal path cover. □

Remark 2.5. Let $S=s_1,s_2,\ldots,s_{|S|}$ and $T=t_1,t_2,\ldots,t_{|T|}$ be the vertices of S and T in the strong ordering of G. Note that a contiguous path P starting with a vertex $s_i \in S$ will be of the form $s_i t_j s_{i+1} t_{j+1} \ldots s_k t_r (s_{k+1})$, where $1 \leqslant i \leqslant k \leqslant |S|$, $1 \leqslant j \leqslant r \leqslant |T|$. This follows from the proof given in [7] for the Hamiltonian path. In other words, P covers $s_i, s_{i+1}, \ldots, s_k$ and $t_j, t_{j+1}, \ldots, t_r$. A similar remark holds for contiguous paths starting from a vertex in T.

Let $P=s_i t_j s_{i-1} t_{j+1} \cdots t_{l-1} s_r (t_l)$ be a path in G. P is said to be *extendable on right*, or simply *extendable*, if P ends with t_l and $t_l s_{r-1} \in E$ or P ends with s_r and $s_r t_l \in E$. We say that a path is a *maximal path* if it is not possible to extend the path on the right. Note that each optimal path cover can be converted into an optimal path cover in which each path is a maximal path. We say that an optimal contiguous path cover $P=P_1,P_2,\ldots,P_k$ is a *maximum optimal path cover* if each P_l covers maximum number of vertices in $V-\{P_1 \cup P_2 \cup \cdots \cup P_{l-1}\}$.

Lemma 2.6. *Let G be a bipartite permutation graph. Then there exists a contiguous path cover for G which is an optimal maximum path cover.*

Proof. Let $P=P_1,P_2,\ldots,P_r$ be an optimal maximal path cover for G. We will convert this path cover into an optimal path cover in which each path is a maximum path. Let $P_k=s_i t_j s_{i-1} \ldots s_r t_l$ be the first path in P which is not a maximum path. Without loss of generality, assume that P_k ends in a t vertex. Then the maximum path P_k' will be of the form $P_k=t_j s_i \ldots t_l s_r t_{l-1} s_{r-1} \ldots t_u (s_r)$. If the path P_{k-1} starts from s_{r+1} then we can replace P_k and P_{k+1} with P_k' and P_{k-1}', where $P_{k-1}'=P_{k+1}-P_k'$. If P_{k+1}' starts from the vertex t_{l+1} then P_k' with replace P_k and P_{k-1}, which will contradict the optimality. By repeating the procedure for all non maximum paths, we can get an optimal path cover in which each path is a maximum path. □

Algorithm 1

(1) Mark all vertices in S and T as not visited. Let $P=\emptyset$.
(2) While all the vertices are not visited Do
  (2.1) Let s and t be the first vertices in S and T which are not visited.
  (2.2) Let P_s and P_t be the maximal paths starting from s and t, respectively. $Q=$ Maximum of P_s and P_t.
  (2.3) $P=P \cup Q$.
  (2.4) Mark all vertices in Q as visited.
  End while.
(3) Output P.

Theorem 2.7. *Algorithm* 1 *outputs an optimal path cover.*

Proof. Follows from Lemma 2.6. □

Complexity. Strong ordering of the given bipartite permutation graph can be found in linear time [7]. Note that each vertex is examined atmost twice. Hence, Algorithm 1 runs in $O(n+m)$ time. For other details, refer to [8].

Theorem 2.8. *The optimal path cover problem can be solved in linear time for the class of bipartite permutation graphs.* □

3. Block graphs

A block of a graph is a maximal biconnected subgraph of the graph. A vertex v of a connected graph G is a cut vertex if $G-v$ is disconnected. A block graph is a graph in which each block is a complete graph. For every block graph $G=(V,E)$, we define its BC-tree T as follows: T has a node corresponding to each block and each cut vertex of G. The nodes of T corresponding to blocks of G are called block nodes and the nodes corresponding to cut vertices of G are called cut nodes. We say, a cut node c is contained in a block node b if the block represented by b contains c. Every cut node c is connected to all block nodes which contain c.

3.1. Optimal path cover in block graphs

It is instructive to note that the minimum path cover number for the graph T need not be equal to the minimum path cover number of G (Fig. 1). We will assume that T is rooted at a cut node. The nodes of T will be visited in post order traversal. For each node $v \in T$, G_v denotes the subgraph of G formed by the vertices in the blocks represented by descendant block nodes of v except for the vertex w, where w is

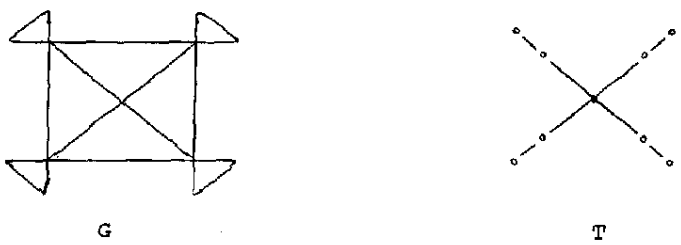

Fig. 1. The optimal path cover number of G is 2. The optimal path cover number of T is 3.

the cut vertex v if v is a cut node or the cut vertex u, and the parent of v if v is a block node.

Algorithm 2

(1) Construct the BC-tree T of G and root the tree T at a cut node.

(2) While the root is not visited Do

(2.1) For all nodes v of T whose children are visited Do

(2.1.1) If v is a cut node then let $b_1, b_2, \ldots, b_k$ be the children of v. Let $P_{b_1}, P_{b_2}, \ldots, P_{b_k}$ be the optimal path covers of $G_{b_1}, G_{b_2}, \ldots, G_{b_k}$, respectively.

If there are paths $P'_{b_i} \in P_{b_i}$ and $P'_{b_j} \in P_{b_j}$ such that paths P'_{b_i} and P'_{b_j} end in c' and c'', respectively, and $c', c \in b_i$ and $c'', c \in b_j$ then $P_v = P_{b_1} \cup P_{b_2} \cup \cdots \cup (P_{b_i} - P'_{b_i}) \cup \cdots \cup (P_{b_j} - P'_{b_j}) \cup \cdots \cup P_{b_k} \cup (P'_{b_i} - c' - c - c'' - P'_{b_j})$. (See Fig. 2)

If there is only one such path $P'_{b_i} \in P_{b_i}$ then $P_v = P_{b_1} \cup P_{b_2} \cup \cdots \cup (P_{b_i} - P'_{b_i}) \cup \cdots \cup P_{b_k} \cup (P'_{b_i} - c' - c)$.

If there is no such path then $P_v = P_{b_1} \cup P_{b_2} \cup \cdots \cup P_{b_i} \cup \cdots \cup P_{b_k} \cup (c)$.

(2.1.2) If v is a block node then,

let $b_1, b_2, \ldots, b_t$ be the non cut vertices in the block represented by the block node v.

If v has only one son c then

If there is a path $P_1 \in P$, where P is the optimal path cover of G_c and if P_1 ends in c

then $P_v = (P - P_1) \cup (P_1 - c - b_1 - b_2 - \cdots - b_t)$

else $P_v = P \cup (b_1 - b_2 - \cdots - b_t)$.

else

Let $c_1, c_2, \ldots, c_k$ be the children of v. Let $P_{c_1}, P_{c_2}, \ldots, P_{c_k}$ be the optimal path covers of $G_{c_1}, G_{c_2}, \ldots, G_{c_k}$, respectively.

Let $P'_{c_{i1}}, P'_{c_{i2}}, \ldots, P'_{c_{ir}}$ be the paths that end in vertices $c_{i1}, c_{i2}, \ldots, c_{ir}$, respectively.

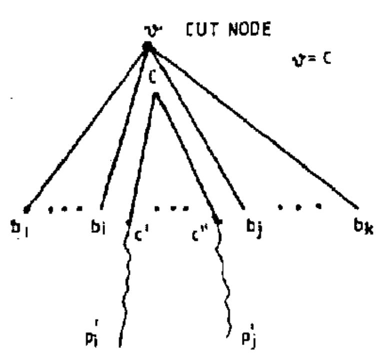

Fig. 2.

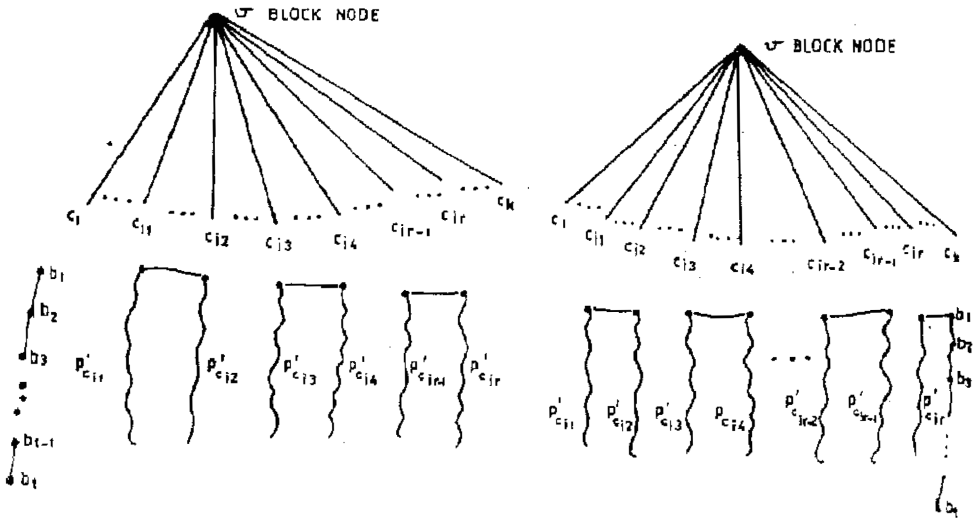

Fig. 3.

Then $P_v = (P'_{c_{i1}} - c_{i1} - c_{i2} - P'_{c_{i2}}) \cup (P'_{c_{i3}} - c_{i3} - c_{i4} - P_{c_{i4}}) \cup \cdots \cup$
$(P'_{c_{iq-1}} - c_{iq-1} - c_{iq} - P'_{c_{iq}}) \cup ((P_{c_{i1}} \cup P_{c_{i2}} \cup \cdots \cup P_{c_{ik}}) - (P'_{c_{i1}},$
$P'_{c_{i2}}, \ldots, P'_{c_{ir}})$ (where $q = r$ or $r-1$, depending upon whether
r is divisible by 2 or not). (See Fig. 3)
If ir is divisible by 2 then $P_v = P_v \cup (b_1 - b_2 - \cdots - b_l)$
else $P_v = P_v \cup (P_{v_{ir}} - b_1 - b_2 - \cdots - b_l)$

Mark v as visited.

End for.

End while.

We can easily prove the correctness of Algorithm 2 by induction on the number of vertices of the given graph [1]. At each step the algorithm processes a node and its children. Hence, Algorithm 2 takes only $O(n+m)$ time to find an optimal path cover.

Theorem 3.1. *The optimal path cover problem can be solved in linear time for the class of block graphs.*

4. Conclusions

We have presented linear algorithms for the path cover problem on bipartite permutation graphs and block graphs. The earlier polynomial-time algorithms have been predominantly using greedy techniques and in our case, for the bipartite permutation graphs, we have used greedy methods while, for the block graphs, we have used the block cut vertex tree as the tool. The block cut vertex tree has been emerging as a paradigm for solving a number of problems in block graphs [5]. While we have been successful in designing polynomial-time algorithms for this problem on interval graphs, block graphs etc., it is well known that minimal path cover problem is NP-complete even for chordal graphs. Thus, it would be interesting to investigate the complexity of the path cover problem on larger classes of graphs and identify a sharper dividing line between P- and NP-completeness of this problem across different classes of graphs.

References

[1] A. Aho, J.E. Hopcroft and J.D. Ullman, *Data Structures and Algorithms* (Addison-Wesley, Reading, MA, 1983).

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

[3] M.R. Gary and D.S. Johnson, *Computers and Intractability: A Guide to the theory of NP-Completeness* (Freeman, San Francisco, CA, 1979).

[4] S. Moran and Y. Wolfstahl, Optimal covering of cacti by vertex disjoint paths, *Theoret. Comput. Sci.* **84** (1991) 179–197.

[5] C.P. Rangan, T.S. Jayaram and G. Srikrishna, A unified approach for solving domination problems on block graphs, Available as TCS-90-02, Department of Computer Science, Indian Institute of Technology, Madras, India, 1990, submitted.

[6] J. Spinrad, Two dimensional partial orders, Ph.D dissertation, Princeton University, Princeton, NJ, 1982.

[7] J. Spinrad, A. Brandstadt and L. Stewart, Bipartite permutation graphs, *Discrete Appl. Math.* **18** (1987) 279–292.

[8] R. Srikant and C.P. Rangan, Linear algorithm for optimal path cover problem on bipartite permutation graphs, Tech. Report TCS-89-15, Department of Computer Science, Indian Institute of Technology, Madras, India, 1989.

[9] A.S. Rao and C.P. Rangan, Linear algorithm for optimal path cover problem on interval graphs, *Inform. Process. Lett.* **35** (1990) 149–153.