Theoretical Computer Science 591 (2015) 99–105

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A linear-time algorithm for paired-domination on circular-arc graphs ☆

Ching-Chi Lin [a,*], Hai-Lun Tu [b]

[a] Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 20224, Taiwan
[b] Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan

ARTICLE INFO

Article history:
Received 9 October 2014
Received in revised form 30 April 2015
Accepted 3 May 2015
Available online 7 May 2015
Communicated by V.Th. Paschos

Keywords:
Paired-domination problem
Perfect matching
Interval graph
Circular-arc graph

ABSTRACT

In a graph G, a vertex subset $S \subseteq V(G)$ is said to be a dominating set of G if every vertex not in S is adjacent to a vertex in S. A dominating set S of a graph G is called a paired-dominating set if the induced subgraph $G[S]$ contains a perfect matching. The paired-domination problem involves finding a minimum paired-dominating set of G. For this problem, Chen et al. [J. Comb. Optim. 19 (4) (2010) 457–470] proposed an $O(n+m)$-time algorithm on interval graphs and Cheng et al. [Discrete Appl. Math. 155 (16) (2007) 2077–2086] designed an $O(m(n+m))$-time algorithm on circular-arc graphs. In this paper, we strengthen the results of Cheng et al. by showing an $O(n+m)$-time algorithm. Moreover, the algorithm can be completed in $O(n)$ time if an intersection model of a circular-arc graph G with sorted endpoints is given. Since interval graphs are circular-arc graphs, we also obtain a linear time algorithm on interval graphs.

1. Introduction

The museum protection problem can be accurately represented by a graph $G = (V, E)$. The vertex set of G, denoted by $V(G)$, represents the sites to be protected; and the edge set of G, denoted by $E(G)$, represents the set of protection capabilities. There exists an edge (x, y) connecting vertices x and y if a guard at site x is capable of protecting site y and vice versa. In the classical domination problem, we are asked to minimize the number of guards such that each site has a guard or is in the protection range of some guard. For the paired-domination problem, in addition to protecting the sites, the guards must be able to back each other up [1]. Throughout this paper, we let $n = |V(G)|$ and $m = |E(G)|$.

In a graph G, a vertex subset $S \subseteq V(G)$ is said to be a *dominating set* of G if every vertex not in S is adjacent to a vertex in S. Let $G[S]$ denote the subgraph of G induced by a subset S of $V(G)$. A dominating set S of a graph G is called a *paired-dominating set* if the induced subgraph $G[S]$ contains a perfect matching. The *paired-domination problem* involves finding a minimum paired-dominating set of G. Haynes and Slater [1] defined the paired-domination problem and showed that it is NP-complete in general graphs. More recently, Chen et al. [2] demonstrated that the problem is also NP-complete in bipartite graphs, chordal graphs, and split graphs. Panda and Pradhan [3] strengthened the above result for bipartite graph by showing that the problem is NP-complete for perfect elimination bipartite graphs. In addition, McCoy and Henning [4] investigated variants of the paired-domination problem in graphs.

☆ This work is partially supported by the National Science Council Taiwan under the Grants No. NSC-101-2221-E-019-065, and NSC-102-2221-E-019-038.
* Corresponding author.
E-mail addresses: lincc@mail.ntou.edu.tw (C.-C. Lin), d95019@csie.ntu.edu.tw (H.-L. Tu).

http://dx.doi.org/10.1016/j.tcs.2015.05.002
0304-3975/

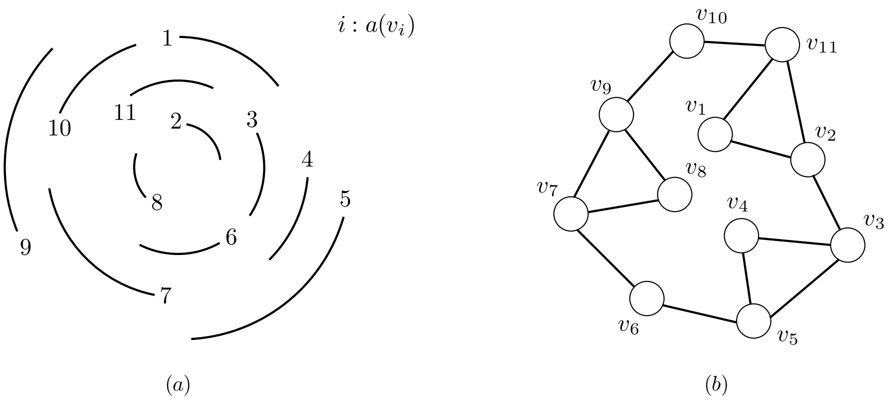

Fig. 1. (a) A family of arcs on a circle. (b) The corresponding circular-arc graph G for the family of arcs in (a).

Furthermore, several polynomial-time algorithms have been developed for some special classes of graphs such as trees, interval graphs, strongly chordal graphs, and circular-arc graphs. Qiao et al. [5] proposed an $O(n)$-time algorithm for trees; Chen et al. [6] improved the above result by providing an $O(n)$-time algorithm for weighted trees; Kang et al. [7] presented an $O(n)$-time algorithm for inflated trees; Chen et al. [6] designed an $O(n+m)$-time algorithm for strongly chordal graphs; and Cheng et al. [8] developed an $O(mn)$-time algorithm for permutation graphs. To improve the results in [8], Lappas et al. [9] introduced an $O(n)$-time algorithm. In addition, Hung [10] described an $O(n)$-time algorithm for convex bipartite graphs; Panda and Pradhan [3] proposed an $O(n+m)$-time algorithm for chordal bipartite graphs; Chen et al. [2] introduced $O(n+m)$-time algorithms for block graphs and interval graphs; and Cheng et al. [11] designed an $O(n+m)$-time algorithm for interval graphs and an $O(m(n+m))$-time algorithm for circular-arc graphs.

It is known that the classes of strongly chordal graphs and circular-arc graphs are proper superclasses of the class of interval graphs. Therefore, to strengthen the linear-time algorithms for interval graphs [2,11], one can either design a linear-time algorithm for strongly chordal graphs [6] or propose a linear-time algorithm for circular-arc graphs. In this paper, we strengthen the result in [2,11] by showing an $O(n+m)$-time algorithm for circular-arc graphs. Moreover, the algorithm can be completed in $O(n)$ time if an intersection model of a circular-arc graph G with sorted endpoints is given. Obviously, as a byproduct, we also obtain a linear time algorithm for interval graphs. Over the past two decades, several variants of the classic domination problem, such as independent domination, connected domination, and power domination, have studied intensively in some special classes of graphs [12–18]. In particular, these problems have been proven to have linear-time algorithms on circular-arc graphs [19–21]. Hence, the results of this paper complete the role of paired-domination problem on circular-arc graphs.

The remainder of this paper is organized as follows. In Section 2, given an intersection model of a circular-arc graph G with sorted endpoints, we present an efficient algorithm for finding a minimum paired-dominating set of G. Section 3 provides an $O(n)$-time implementation of the algorithm proposed in Section 2. Section 4 contains some concluding remarks and future work.

2. The proposed algorithm for circular-arc graphs

A graph G is considered as a *circular-arc graph* if there is a one-to-one correspondence between $V(G)$ and a family of arcs on a circle such that $(u, v) \in E(G)$ if and only if the corresponding arc of u overlaps with the corresponding arc of v. Moreover, an *intersection model* F of a circular-arc graph G is a circular ordering of its corresponding arc endpoints when moving in a clockwise direction around the circle. McConnell [22] proposed an $O(n+m)$-time algorithm that recognize a circular-arc graph G, and simultaneously obtains an intersection model F of G as a byproduct. In the following discussion, we assume that (1) G is a circular-arc graph such that $V(G) = \{v_1, v_2, \ldots, v_n\}$ with $n \geq 3$; and (2) the intersection model F of G is available.

For each $v \in V(G)$, let $a(v)$ denote the corresponding arc of v in F. Each arc is represented by $[h(v), t(v)]$, where $h(v)$ is the *head* of $a(v)$, $t(v)$ is the *tail* of $a(v)$, and $h(v)$ precedes $t(v)$ in a clockwise direction. Furthermore, for any subset R of $V(G)$, we define $a(R) = \{a(v) \mid v \in R\}$. Since $h(v)$ and $t(v)$ are endpoints on a circle, it is assumed that all $h(v)$ and $t(v)$ are distinct and no single arc in F covers the whole circle. All endpoints are assigned positive integers between 1 and $2n$ in ascending order in a clockwise direction.

In addition, we assume that $h(v_1) = 1$. We also assume that $a(v_1)$ is chosen arbitrarily from F; and let $a(v_2), a(v_3), \ldots, a(v_n)$ be the ordering of arcs in $F \setminus \{a(v_1)\}$ such that $h(v_i)$ is encountered before $h(v_j)$ in a clockwise direction from $h(v_1)$ if $i < j$. Fig. 1 shows an illustrative example, in which Fig. 1(b) depicts the corresponding circular-arc graph G for the family of arcs in Fig. 1(a). An ordering of the family of arcs is also provided. For each $v \in V(G)$, let the *tail partner* of v, denoted by $p_t(v)$, be the neighbor of v such that $a(p_t(v))$ contains $t(v)$, and $t(p_t(v))$ is the last tail encountered in clockwise direction from $t(v)$ in F. Similarly, let the *head partner* of v, denoted by $p_h(v)$, be the neighbor of v such that $a(p_h(v))$