Faster Algorithms for the Paired Domination Problem on Interval and Circular-Arc Graphs

Leonidas Palios

Department of Computer Science and Engineering, University of Ioannina

GR-45110 Ioannina, Greece

palios@cs.uoi.gr

Abstract

A vertex subset D of a graph G is a dominating set if every vertex of G is either in D or is adjacent to a vertex in D. The paired domination problem on G asks for a minimum-cardinality dominating set S of G such that the subgraph induced by S contains a perfect matching; motivation for this problem comes from the interest in finding a small number of locations to place pairs of mutually visible guards so that the entire set of guards monitors a given area. The paired domination problem on general graphs is known to be NP-complete.

In this paper, we consider the paired domination problem on interval and circular-arc graphs. We use properties of the models of interval and circular-arc graphs in order to describe simple and efficient algorithms for the problem: given an interval (arc, resp.) model of an interval (circular-arc, resp.) graph on n vertices and m edges with endpoints sorted, our algorithms detect whether there exist isolated vertices, returning one if one exists, otherwise returning a minimum paired-dominating set of the input graph; our algorithm for interval graphs runs in $O(n)$ time and space whereas the one for circular arc graphs runs in $O(n + m)$ time using $O(n)$ space. Both algorithms achieve better time complexities over the corresponding known algorithms.

Keywords: interval graph, circular-arc graph, paired domination, certifying algorithm, domination.

1 Introduction

A subset D of vertices of a graph G is a *dominating set* if every vertex of G either belongs to D or is adjacent to a vertex in D; the minimum cardinality of a dominating set of G is called the *domination number* of G and is denoted by $\gamma(G)$. The problem of computing the domination number of a graph has received and keeps receiving considerable attention by many researchers (see [15] for a long bibliography on domination). The problem finds many applications, most notably in relation to area monitoring problems by a minimum-cardinality set of guards.

The domination problem admits many variants: domination, edge domination, weighted domination, independent domination, connected domination, total/open domination, locating domination, and paired domination [15, 16, 17, 18, 26, 32]. Among these, we will focus on paired domination: a vertex subset S of a graph G is a *paired-dominating set* if it is a dominating set and the subgraph induced by the set S has a perfect matching; the minimum cardinality of a paired-dominating set in G is called the *paired domination number* and is denoted by $\gamma_p(G)$. Paired domination was introduced by Haynes and Slater [17]; their motivation came from the variant of the area monitoring problem in which each guard has another guard as a backup (i.e., we have pairs of guards protecting each other). Haynes and Slater noted that every graph with no isolated vertices has a paired-dominating set (on the other hand, it easily follows from the definition that a graph with isolated vertices does not have a paired-dominating set).

Additionally, they showed that the paired domination problem is NP-complete on arbitrary graphs; thus, it is of theoretical and practical importance to find classes of graphs for which this problem can be solved in polynomial time and to describe efficient algorithms for its solution.

Trees have been one of the first targets of researchers working on paired domination: Qiao *et al.* [28] presented a linear-time algorithm for computing the paired domination number of a tree and characterized the trees with equal domination and paired domination number; Henning and Plummer [19] characterized the set of vertices of a tree that are contained in all, or in no minimum paired-dominating sets of the tree. Kang *et al.* [21] considered "inflated" graphs (for a graph G, its inflated version is obtained from G by replacing each vertex of degree d in G by a clique on d vertices), gave an upper and lower bound for the paired domination number of the inflated version of a graph, and described an algorithm for computing a minimum paired-dominating set of the inflated version of a tree T_r which runs in $O(|V(T_r)|)$ time. Bounds for the paired domination number have been established also for claw-free cubic graphs [12], for Cartesian products of graphs [3], and for generalized claw-free graphs [9]; we call $K_{1,3}$ a claw and $K_{1,a}$ a generalized claw, where $a \geq 3$, and thus a graph G is called claw-free (generalized claw-free, resp.) graph if G does not contain $K_{1,3}$ ($K_{1,a}$, resp.) as an induced subgraph. Cheng *et al.* [8] gave an $O(nm)$-time algorithm for the paired domination problem on permutation graphs, where n and m are the numbers of vertices and edges of the graph, working on the permutation defining the input graph; an optimal $O(n)$-time algorithm for this problem was recently described by Lappas *et al.* [25]. For the paired domination problem on interval graphs, Cheng *et al.* [7] proposed an $O(n + m)$-time algorithm assuming that an interval model for the graph with endpoints sorted is available; they also extended their result to circular-arc graphs giving an algorithm running in $O(m(m + n))$ time in this case. Chen *et al.* [6] pointed out that the interval graph algorithm in [7] is incorrect and gave $O(n + m)$-time algorithms for the paired domination problem on block graphs provided that an appropriate vertex ordering is given and on interval graphs provided that an interval model with endpoints sorted is given; they also showed that the problem is NP-complete for bipartite, chordal, and split graphs. The same authors....

Chen *et al.* [5] described an $O(n + m)$-time algorithm for the paired domination problem on strongly chordal graphs if the strong (elimination) vertex ordering is given; their algorithm implies an $O(n + m)$-time algorithm for the paired-domination problem on interval graphs when

We too consider the paired domination problem on the classes of interval and circular-arc graphs. An *interval graph* is the intersection graph of a family of intervals in the real line; the class of interval graphs is a subclass of the very interesting class of perfect graphs [13]. Recognizing whether a graph on n vertices and m edges is interval can be done in $O(n+m)$ time [2, 24, 14]; in fact, the algorithms in [24] and [14] produce an *interval model* whenever the input graph is found to be interval. The circular-arc graphs generalize the interval graphs; a *circular-arc graph* is the intersection graph of a family of arcs on a circle. McConnell [27] gave an $O(n + m)$-time algorithm to recognize whether a given graph is circular-arc. In 2006, Kaplan and Nussbaum [22] described a simpler $O(n + m)$-time circular-arc graph recognition algorithm based on an earlier $O(n^2)$-time algorithm of Eschen and Spinrad [10]. Both the algorithms of McConnell and of Kaplan and Nussbaum produce a corresponding *arc model* if the given graph is circular-arc graph.

Both the interval and the circular-arc graphs have received considerable attention and many algorithms have been developed for various problems on these graphs. In addition to the result of Cheng, Kang, and Ng [7] on paired domination that we mentioned earlier, several variants of the domination problem have been considered on interval and circular-arc graphs. Farber [11] presented a polynomial-time algorithm for computing a minimum-weight dominating set and a minimum-weight independent dominating set on strongly chordal graphs that require $O(n + m)$ time on interval graphs. White *et al.* [31] gave an $O(n^2)$-time algorithm for a minimum-cardinality connected dominating set for strongly chordal graphs and thus for interval graphs. Bertossi [1] described an $O(n^2)$-time algorithm for computing a minimum-cardinality total dominating set on an interval graph. The same year, Keil [23] proposed an improved algorithm for the same problem that run in $O(n+m)$ time; Ramalingam and Pandu Rangan [29] pointed out an error in Keil's algorithm and corrected it. The same authors in [30] described a unified approach leading to $O(n+m)$-time algorithms for the minimum-weight versions of the domination, independent domination, total domination, and connected domination on interval graphs. In 1998, Chang [4] gave $O(n)$-time algorithms for minimum-weight {independent, connected} domination, and an $O(n \log\log n)$-time algorithm for minimum-weight total domination on interval graphs assuming that an interval model with endpoints sorted is given; he also extended the results to circular-arc graphs obtaining $O(n+m)$-time algorithms for the same problems. We also note that Hsu and Tsai [20] presented an $O(n)$-time algorithm for the minimum-cardinality dominating set (as well as the minimum independent set and the minimum clique cover) on circular-arc graphs assuming that an arc model is given.

In this paper, we study the paired domination problem on interval and circular-arc graphs, assuming that an interval and arc representation of the graph with endpoints sorted is given. We prove properties of the intervals and the arcs in the representation which help us describe an optimal $O(n)$-time algorithm for the paired-dominating problem on interval graphs and an $O(n+m)$-time algorithm for the circular-arc graphs. Since an interval model of an interval graph and an arc model of a circular-arc graph can be computed in time linear in the total number of vertices and edges of the graph, our algorithms imply $O(n+m)$-time algorithms for interval and circular-arc graphs when the graph is given.

2 Theoretical Framework

We consider finite undirected graphs with no loops or multiple edges. For a graph G, we denote its vertex and edge set by $V(G)$ and $E(G)$, respectively. The subgraph of G induced by a subset S of the vertex set $V(G)$ is denoted by $G[S]$. The *neighborhood* $N(x)$ of a vertex x of G is the set of all the vertices of G which are adjacent to x; the *closed neighborhood* of x is defined as $N[x] := N(x) \cup \{x\}$. The *degree* of a vertex x in G is the number of vertices adjacent to x in G; thus, $degree(x) = |N(x)|$.

Our algorithms assume that an interval model of an interval graph and an arc model for a circular-arc graph is given with endpoints sorted. Furthermore, for convenience, we assume that the intervals and the arcs have distinct endpoints. Yet, even if we had a model in which intervals or arcs may have the same endpoint, then we can easily get a model with distinct endpoints as follows: first, to each vertex v of the graph, we arbitrarily assign a distinct integer from 1 to n, denoted $id(v)$, where n is the number of vertices of the graph; then, an endpoint of the interval or the arc of a vertex w at $x = x_i$ is represented by the ordered pair $(x_i, id(w))$ and the comparison of the endpoints is done lexicographically on the corresponding ordered pairs. This corresponds to moving the endpoint of the interval (arc, resp.) corresponding to the larger id a bit to the right (clockwise, resp.).

3 Paired Domination of Interval Graphs

In this section, we present and analyze the algorithm for the paired domination problem on interval graphs; we assume that an interval model with endpoints sorted is given.

The general idea of our algorithm is to traverse the intervals in the interval model of the input graph from left to right

- collecting pairs of adjacent vertices whose intervals extend as far to the right as possible
- without however leaving behind intervals corresponding to non-dominated vertices.

This can be done in a systematic way by taking advantage of the result described in the following lemma:

Lemma 3.1 *Let $\mathcal{I}_G$ be an interval model of an interval graph G without isolated vertices and let*
v_i be the non-dominated vertex of G whose interval in $\mathcal{I}_G$ has the leftmost *right endpoint,*
v_j be the neighbor of v_i whose interval in $\mathcal{I}_G$ has the rightmost *right endpoint, and*

v_k be the neighbor of v_j whose interval in $\mathcal{I}_G$ has the rightmost *right endpoint. Then there exists a minimum paired-dominating set of G which contains the pair $\{v_j, v_k\}$.*

Proof: Since the graph G has no isolated vertices, the vertices v_j and v_k exist (note that it may hold that $v_k = v_i$). Consider a minimum paired-dominating set S of G. First, we show that S contains a neighbor of v_i. If not, then $v_i \in S$; since the induced subgraph $G[S]$ has a perfect matching, v_i is matched to one of its neighbors belonging to S, a contradiction. Thus, S contains a neighbor of v_i. If S does not contain v_j, then we can obtain a minimum paired-dominating set S' of the paired domination problem on G containing v_j by simply replacing a neighbor of v_i in S by v_j; note that the definitions of v_i and v_j imply that v_j is adjacent to all the neighbors of v_i and of v_i's neighbors.

Next, since S' is a minimum paired-dominating set of G, the subgraph $G[S']$ of G induced by S' has a perfect matching; thus v_j is matched to another vertex in S', say, w. If $w = v_k$ then S' contains both v_j and v_k. If $w \neq v_k$, then we can obtain a minimum paired-dominating set containing both v_j and v_k by replacing w by v_k; v_j dominates all the vertices whose intervals in $\mathcal{I}_G$ start to the left of the left endpoint of the interval of v_j while the definition of v_k implies that for any neighbor w of v_j it holds that $N(w) - N(v_j) \subseteq N(v_k) - N(v_j)$. ∎

Let us denote by $I(v)$ the interval corresponding to vertex v in an interval model. In order to simplify our presentation, let us denote by $r\_neighbor(v)$ the neighbor of vertex v whose interval in the interval model has the rightmost right endpoint; thus, for a vertex v, $r\_neighbor(v)$ is well defined as long as v is not an isolated vertex. We note that if the intervals of the neighbors of v do not extend past the right endpoint of the interval $I(v)$ of v, the right endpoint of the interval of $r\_neighbor(v)$ will be *to the left* of the right endpoint of $I(v)$.

Then, our method to compute a paired-dominating set of an interval graph G with interval model $\mathcal{I}_G$, as suggested by Lemma 3.1, is as follows: we initialize the dominating set of G to the empty set; next, we find the vertex, say, v, whose interval in $\mathcal{I}_G$ has the leftmost right endpoint and we add the doubleton set $\{r\_neighbor(v),\ r\_neighbor(r\_neighbor(v))\}$ in the dominating set of G; following that, we ignore all the vertices dominated by the current dominating set and find the vertex, say, v', (among the vertices that are not yet dominated) whose interval in $\mathcal{I}_G$ has the leftmost right endpoint and we add the doubleton set $\{r\_neighbor(v'),\ r\_neighbor(r\_neighbor(v'))\}$ in the dominating set of G; we keep repeating the last step for as long as there are non-dominated vertices.

It is interesting to note that the choice of pairs of adjacent vertices guarantees that at any time, the interval of any non-dominated vertex v starts to the right of the intervals of all the vertices in the current dominating set. This implies that $r\_neighbor(v)$ does not belong to the current dominating set, nor does $r\_neighbor(r\_neighbor(v))$.

Of course, if there exist isolated vertices in the graph G, the paired domination problem on G has no solution [17]. So, in its Step 1, our algorithm checks for isolated vertices and computes the values of $r\_neighbor(x)$ for all vertices $x \in V(G)$. If isolated vertices are found, an appropriate message is printed and the algorithm stops, whereas if no such vertices exist our algorithm applies the method described in the previous paragraph. A description of our algorithm in pseudocode is given in Algorithm Interval_Paired_Domination.

Algorithm Interval_Paired_Domination($\mathcal{I}_G$)
Input : an interval model $\mathcal{I}_G$ of an interval graph G with interval endpoints sorted
Output : a minimum paired-dominating set of G, if it exists, or
a message that there is no solution and an isolated vertex of G

---

1. traverse the interval endpoints in $\mathcal{I}_G$ (from left to right) in order to check for isolated vertices and to compute the value of $r\_neighbor(v)$ $\forall v \in V(G)$;
   if there exists a vertex w that is isolated in G
   then print("No solution...");
   print that w is an isolated vertex and **exit**;

2. $\{$*traverse interval endpoints again (from left to right) to get a minimum paired-dominating set*$\}$
   mark all vertices in G with -1; $\{$*-1 denotes not yet encountered vertex*$\}$
   $S \leftarrow \emptyset$; $\{$*S will store a dominating set; initially empty*$\}$
   $i \leftarrow 0$; $\{$*counter for pairs in S; initially 0 pairs*$\}$
   while there exist interval endpoints to be processed **do**
   $p \leftarrow$ next interval endpoint in $\mathcal{I}_G$;
   $v \leftarrow$ vertex corresponding to the interval with p as an endpoint;
   if p is the left endpoint of $I(v)$
   then mark v with i; $\{$*$I(v)$ encountered (v non-dominated) after the i-th pair in S*$\}$
   else $\{$*p is the right endpoint of $I(v)$*$\}$
   if v is marked with i
   then $\{$*v: non-dominated vertex whose right endpoint is leftmost*$\}$
   $S \leftarrow S \cup \{r\_neighbor(v), r\_neighbor(r\_neighbor(v))\}$;
   skip endpoints in $\mathcal{I}_G$ up to the rightmost between the right endpoints of $I(r\_neighbor(v))$ and $I(r\_neighbor(r\_neighbor(v)))$;
   $i \leftarrow i + 1$; $\{$*increment counter for next pair in S*$\}$
   end-while

3. **print**("A minimum paired-dominating set of the input graph is:");
   print the elements of the set S.

The correctness of Algorithm Interval_Paired_Domination follows from Lemma 3.1 and the discussion preceding the pseudocode. Additionally, as the set S is constructed by collecting disjoint pairs of adjacent vertices, this guarantees that the subgraph of the input graph G induced by the resulting set S will have a perfect matching.

Time and Space Complexity

Let n be the numbers of vertices of the given graph G. In order to achieve a good time complexity, we establish pointers from each endpoint of each interval $I(v)$ to the corresponding vertex v and with each vertex we store the values of the endpoints of its corresponding interval; these can be set in $O(n)$ time by means of an initial traversal of the intervals in the interval model $\mathcal{I}_G$. Then, Step 2 runs in in time linear in the number of interval endpoints, that is, in $O(n)$ time and uses $O(n)$ space. Step 3 also takes $O(n)$ time.

Let us now see how we can implement Step 1 in $O(n)$ time and space as well. The computation of *r_neighbor*s relies in maintaining the value of *rightmost_v*, i.e., the vertex whose interval has the rightmost right endpoint so far. Then, $r\_neighbor(x)$ is equal to the value of *rightmost_v* when the right endpoint of the interval $I(x)$ of x is reached unless it happens that the value of *rightmost_v* is equal to x. The latter holds if and only if none of the intervals of the neighbors of x extends past the right endpoint of $I(x)$; in such a case, the $r\_neighbor(v)$ is the vertex whose interval ended last before the right endpoint of $I(x)$ was reached (maintained in *previous_v* in our algorithm) provided that x has neighbors. If x has neighbors then *previous_v* differs from x and is indeed $r\_neighbor(x)$. If x has no neighbors (i.e., it is an isolated vertex) then no interval endpoint appears between the endpoints of $I(x)$ in the interval model, i.e., *previous_v* is equal to x; we take advantage of precisely this observation in order to detect isolated vertices. Below, we present the implementation of Step 1 in pseudocode:

while there exist interval endpoints to be processed **do**
$p \leftarrow$ next interval endpoint in $\mathcal{I}_G$;
$v \leftarrow$ vertex corresponding to the interval with p as an endpoint;
if p is the left endpoint of $I(v)$
then if p is the leftmost interval endpoint in $\mathcal{I}_G$ **or**

the right endpoint of $I(v)$ is to the right of the right endpoint of $I(rightmost\_v)$
then $rightmost\_v \leftarrow v$; {*the right endpoint of* $I(v)$ *is currently rightmost*}
else {p *is the right endpoint of* $I(v)$}
if $rightmost\_v = v$
then {*the intervals of* v*'s neighbors do not extend past the right endpoint of* $I(v)$}
if $previous\_v = v$
then vertex v is an *isolated* vertex of the input graph;
exit from the while-loop;
else $r\_neighbor(v) \leftarrow previous\_v$; {$previous\_v$ *is* v*'s* $r\_neighbor$}
else $r\_neighbor(v) \leftarrow rightmost\_v$; {*set* $r\_neighbor(v)$}
$previous\_v \leftarrow v$;
end-while

In summary, we have the following theorem.

Theorem 3.1 *Let G be an interval graph on n vertices. Then, given an interval model of G with endpoints sorted from left to right, Algorithm* INTERVAL_PAIRED_DOMINATION *computes a paired-dominating set of G in $O(n)$ time and space.*

Since an interval model corresponding to an interval graph can be computed from the graph in time linear in the total number of its vertices and edges (e.g., [24, 14]), we conclude that, given an interval graph, we can compute a minimum-cardinality paired-dominating set of the graph in $O(n + m)$ time, where n is the number of vertices and m is the number of edges of the graph.

4 Paired Domination of Circular-Arc Graphs

In this section, we present and analyze the algorithm for the paired domination problem on circular-arc graphs; we assume that we are given an arc model of the input circular-arc graph with endpoints sorted (recall that we assume that the arcs have distinct endpoints).

Since we have an optimal algorithm for the paired domination problem on interval graphs when given an interval model, it is worth trying to reduce the problem on circular-arc graphs into that on interval graphs. This can be easily done whenever the arc model of the input circular-arc graph G has a *gap*, that is, the union of angle ranges of the arcs in the model do not span the full range of 360 degrees; in such a case, we can obtain an interval model of G by "unrolling" the arcs of the arc model of the circular-arc graph onto a line and then use Algorithm INTERVAL_PAIRED_DOMINATION on it. If the arc model has no gap, then we are able to consider subgraphs of the given graph whose arc models have gaps and reduce again the problem to that on interval graphs.

In order to make our description more precise, we need some additional terminology and notation, which are introduced in Section 4.1; the theoretical background of our algorithm is given in Section 4.2, and the algorithm in Section 4.3.

4.1 Circular-arc Model Terminology and Notation

In an arc model, the arc corresponding to vertex x is denoted by $A(x)$. Each such arc has a *ccw_endpoint* and a *cw_endpoint* and the arc extends in a clockwise direction from the former to the latter and in a counterclockwise direction from the latter to the former (in Figure 1(a), a and b are the ccw_endpoint and cw_endpoint, respectively, of the arc $A(x)$). With respect to the arc of a vertex x, the arc of a neighbor y of x may be such that:

(i) the arc of x *covers* the arc of y (see Figure 1(a));

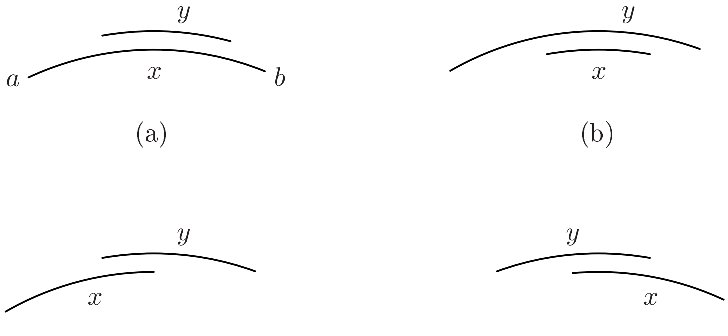

Figure 1: The cases of the arcs of two neighbors x and y.

(ii) the arc of y *covers* the arc of x (see Figure 1(b)) or equivalently the arc of x *is covered* by the arc of y;

(iii) the arcs of x and y *overlap* and the arc of y extends past the cw_endpoint of the arc of x (see Figure 1(c));

(iv) the arcs of x and y *overlap* and the arc of y extends past the ccw_endpoint of the arc of x (see Figure 1(d)).

In cases (i) and (ii) above, we say that x and y form a *nested pair*; in cases (iii) and (iv), they form an *overlapping pair*. In particular, in case (iii) we say that x forms a *clockwise overlapping pair* with y, whereas in case (iv) it forms a *counterclockwise overlapping pair* with y; clearly, if vertex x forms a clockwise overlapping pair with y then y forms a counterclockwise overlapping pair with x, and vice versa.

For a vertex x, the set of neighbors of x can be partitioned into the following 4 sets:

- $N_{cw}(x)$: set of neighbors y of x such that x forms a clockwise overlapping pair with y;
- $N_{ccw}(x)$: set of neighbors y of x such that x forms a counterclockwise overlapping pair with y;
- $N_{covering}(x)$: set of neighbors of x whose arcs cover the arc of x;
- $N_{covered}(x)$: set of neighbors of x whose arcs are covered by the arc of x.

(Note that this partition of the neighbors of x depends on the arc model considered; a different arc model for the same input graph may yield different neighborhood partitions.) Among the elements of $N_{cw}(x)$, $N_{ccw}(x)$, and $N_{covering}(x)$, whenever these sets are non-empty, we distinguish the following special neighbors of x:

- $cw_o(x)$: among the elements of $N_{cw}(x)$ (if any), $cw_o(x)$ is the vertex whose arc extends farthest clockwise;
- $ccw_o(x)$: among the elements of $N_{cw}(x)$ (if any), $ccw_o(x)$ is the vertex whose arc extends farthest counterclockwise;
- $cw_c(x)$: among the elements of $N_{covering}(x)$ (if any), $cw_c(x)$ is the vertex whose arc extends farthest clockwise;
- $ccw_c(x)$: among the elements of $N_{covering}(x)$ (if any), $ccw_c(x)$ is the vertex whose arc extends farthest counterclockwise.

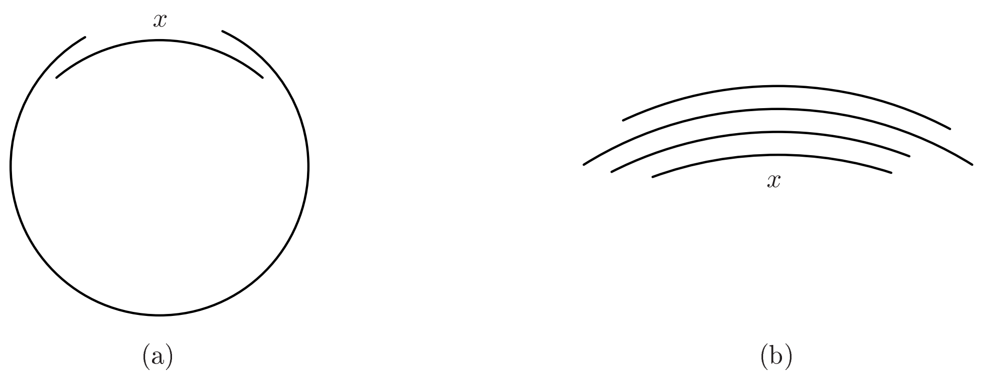

Figure 2: Examples of arcs.

The above vertices are well defined provided that the corresponding set of neighbors of x is non-empty (recall that we assume that the endpoints are all distinct). Moreover, we note that it may hold that $cw_o(x) = ccw_o(x)$ (see Figure 2(a)), as well as $cw_c(x) = ccw_c(x)$ (see Figure 2(b)). It is clear that for a vertex, some or all of the above neighbors need not exist. However, under certain conditions some of these neighbors exist as we show in the following observation.

Observation 4.1 *Let G be a circular-arc graph with arc model $\mathcal{A}_G$. Then:*

(i) For each vertex x whose arc $A(x)$ is covered by another arc in $\mathcal{A}_G$, both neighbors $cw_c(x)$ and $ccw_c(x)$ exist and their arcs are not covered by any arc in $\mathcal{A}_G$.

(ii) If the arc model $\mathcal{A}_G$ does not have a gap, then for each vertex x whose arc $A(x)$ is not covered by any arc in $\mathcal{A}_G$, both neighbors $cw_o(x)$ and $ccw_o(x)$ exist. Moreover, if for a vertex x the neighbor $cw_o(x)$ ($ccw_o(x)$, resp.) exists, then the arc of $cw_o(x)$ ($ccw_o(x)$, resp.) is not covered by any arc in $\mathcal{A}_G$.

(iii) Consider any vertex x whose arc $A(x)$ is not covered by any arc in $\mathcal{A}_G$. If neighbor $y = cw_o(x)$ exists, then $cw_o(ccw_o(y)) = y$. Symmetrically, if neighbor $z = ccw_o(x)$ exists, then $ccw_o(cw_o(z)) = z$.

Proof: (i) Any arc covering the arc $A(x)$ of x extends both clockwise and counterclockwise farther than the endpoints of $A(x)$. Hence both $cw_c(x)$ and $ccw_c(x)$ exist. Moreover, the arc of $cw_c(x)$ is not covered by any arc in $\mathcal{A}_G$; if not, any such arc would cover the arc of x and would extend farther clockwise than the arc of $cw_c(x)$ contradicting the definition of $cw_c(x)$. A similar argument establishes that the arc of $ccw_c(x)$ is not covered by any arc either.

(ii) Since $\mathcal{A}_G$ does not have a gap, there must be an arc extending farther clockwise than the cw_endpoint of the arc of x. The vertex corresponding to this arc is a neighbor of x and belongs to $N_{cw}(x)$ since $N_{covering}(x) = \emptyset$; thus, $cw_o(x)$ exists. Additionally, the arc of $cw_o(x)$ is not covered by any other arc; if there were such an arc $A(w)$ of a vertex w, then $w \in N_{covering}(x) \cup N_{cw}(x) = N_{cw}(x)$, in contradiction to the definition of $cw_o(x)$.
A similar argument holds for $ccw_o(x)$ as well.

(iii) Suppose that $y = cw_o(x)$ exists. Since x forms a clockwise overlapping pair with y, y forms a counterclockwise overlapping pair with x. Thus, $x \in N_{ccw}(y)$ and the vertex $ccw_o(y)$ exists. In turn, $y \in N_{cw}(ccw_o(y))$ and thus $cw_o(ccw_o(y))$ exists. Since the arc of x is not covered by any arc, it is important to note that $N_{ccw}(y)$ contains

- vertices $V_1(y)$ (if any) whose arcs have their cw_endpoints in $A(y) - A(x)$ and their ccw_endpoints in $A(x) - A(y)$,
- vertices $V_2(y)$ whose arcs have their cw_endpoints in $A(x) \cap A(y)$ and their ccw_endpoints in $A(x) - A(y)$, and

- vertices $V_3(y)$ (if any) that belong to $N_{ccw}(x)$ and whose arcs have their cw_endpoints in $A(x) \cap A(y)$.

Then, $ccw_o(y) \in \{x\} \cup V_3(y)$ which implies that $y \in N_{cw}(ccw_o(y))$. If $V_3(y) = \emptyset$, then $ccw_o(y) = x$ and thus $cw_o(ccw_o(y)) = y$. If $V_3(y) \neq \emptyset$, then if $cw_o(ccw_o(y)) = y' \neq y$ (i.e., the arc of y' extends farther clockwise than the cw_endpoint of the arc of y), we have that the ccw_endpoint of y' (i) either belongs to $A(x) \cap A(y)$ which implies that $y' \in N_{cw}(x)$ in contradiction to the definition of $y = cw_o(x)$ (ii) or belongs to $A(x) - A(y)$ which contradicts the fact that the arc of y is not covered by any arc (see statement (ii) for $y = cw_o(x)$). ∎

4.2 Useful Lemmas

Now we are ready to prove the two main lemmas which are the basis of our algorithm. Before that, we show the following fact. We consider a circular-arc graph G whose arc model does not have a gap; thus G has no isolated vertices and there exists a paired-dominating set of G.

Fact 4.1 *Let S be a minimum paired-dominating set of a circular-arc graph G with arc model $\mathcal{A}_G$ that does not have a gap, and let vertices $x, y \in S$ such that x is matched to y in a perfect matching M of the induced subgraph $G[S]$.*

(i) If vertex x forms a clockwise overlapping pair with vertex y in $\mathcal{A}_G$, then there exists a minimum paired-dominating set T of G and perfect matching M_T of the induced subgraph $G[T]$ such that $x, cw_o(x) \in T$ and x is matched to $cw_o(x)$ in M_T.

(ii) If vertex x forms a counterclockwise overlapping pair with vertex y in the arc model $\mathcal{A}_G$, then there exists a minimum paired-dominating set T of G and perfect matching M_T of the induced subgraph $G[T]$ such that $x, ccw_o(x) \in T$ and x is matched to $ccw_o(x)$ in M_T.

(iii) If the arc of vertex x covers the arc of vertex y in the arc model $\mathcal{A}_G$ that does not have a gap, then there exists a minimum paired-dominating set T of G and perfect matching M_T of the induced subgraph $G[T]$ such that $x, cw_o(x) \in T$ and x is matched to $cw_o(x)$ in M_T.

Proof: (i) We first observe that since vertex x forms a clockwise overlapping pair with y, then $cw_o(x)$ exists and

P1: $N[y] - N(x) \subseteq N[cw_o(x)] - N(x)$.

If $cw_o(x) \notin S$ then in light of Property P1, we can obtain a minimum paired-dominating set T as suggested in the statement of the fact by simply replacing y by $cw_o(x)$. So, next suppose that $cw_o(x) \in S$. If $y = cw_o(x)$ then $T = S$. If $y \neq cw_o(x)$ and $cw_o(x) \in S$ then let u be the vertex matched to $cw_o(x)$ in the matching M. Vertex u dominates a vertex w not dominated by any other vertex in S, otherwise the set $S - \{y, w\}$ would also be a paired-dominating set of G due to Property P1, in contradiction to the minimality of S. Then we can obtain a minimum paired-dominating set T as suggested in the statement of the fact by replacing the vertex y by the vertex w in S; note that a perfect matching of the subgraph $G[T]$ is obtained from M by replacing the pairs $\{x, y\}$ and $\{cw_o(x), u\}$ by the pairs $\{x, cw_o(x)\}$ and $\{u, w\}$.
(ii), (iii) Statements (ii) and (iii) are established in a similar fashion. The existence of vertex $cw_o(x)$ in statement (iii) follows from statement (ii) of Observation 4.1 since the arc model $\mathcal{A}_G$ does not have a gap and the arc of x in $\mathcal{A}_G$ is not covered by any arc. ∎

Lemma 4.1 *Let S be a minimum paired-dominating set of a circular-arc graph G with arc model $\mathcal{A}_G$ that does not have a gap, and let $x \in S$. Then:*

(i) If the arc of x is covered *by another arc in $\mathcal{A}_G$, then there exists a minimum paired-dominating set of G containing $cw_c(x)$;*

(ii) If the arc of x is not covered *by another arc in $\mathcal{A}_G$, then there exists a minimum paired-dominating set D of G and perfect matching M_D of the induced subgraph $G[D]$ such that*

- ▷ $ccw_o(cw_o(x)), cw_o(x) \in D$ *and* $ccw_o(cw_o(x))$ *and* $cw_o(x)$ *are matched in* M_D *or*
- ▷ $ccw_o(x), cw_o(ccw_o(x)) \in D$ *and* $ccw_o(x)$ *and* $cw_o(ccw_o(x))$ *are matched in* M_D

where none of the arcs of $cw_o(x)$, $ccw_o(x)$, $ccw_o(cw_o(x))$, and $cw_o(ccw_o(x))$ is covered by any arc in $\mathcal{A}_G$.

Proof: (i) Since the arc of x is covered, then the vertex $cw_c(x)$ exists (as does $ccw_c(x)$). If $cw_c(x) \in S$ then S is a paired-dominating set as described in statement (i); if not, then we can replace x by $cw_c(x)$ in S and obtain such a paired-dominating set since $N[x] \subseteq N[cw_c(x)]$.

(ii) Since the arc of x is not covered by any arc, the vertices $cw_o(x)$ and $ccw_o(x)$ exist and since the arcs of these vertices are not covered either (see statement (ii) of Observation 4.1), then the vertices $ccw_o(cw_o(x))$ and $cw_o(ccw_o(x))$ exist as well. Let $y \in S$ be the vertex matched to x in a perfect matching M of the subgraph $G[S]$. Then, exactly one of the following 3 cases holds:

- x forms a *clockwise overlapping pair* with y in the arc model $\mathcal{A}_G$;
- x forms a *counterclockwise overlapping pair* with y in $\mathcal{A}_G$;
- the arc of x *covers* the arc of y in $\mathcal{A}_G$.

These 3 cases correspond to statements (i), (ii), and (iii), respectively, of Fact 4.1, which implies that in all cases there exists a minimum paired-dominating set T of G and perfect matching M_T of the induced subgraph $G[T]$ such that

$$x, cw_o(x) \in T \text{ and } x \text{ is matched to } cw_o(x) \text{ in } M_T \quad \text{or}$$
$$x, ccw_o(x) \in T \text{ and } x \text{ is matched to } ccw_o(x) \text{ in } M_T.$$

Then, statement (ii) of the lemma follows from once again applying statement (i) of Fact 4.1 in the former case (with respect to $cw_o(x)$) and statement (ii) of Fact 4.1 in the latter case (with respect to $ccw_o(x)$). Note that since the arc of x is not covered by any arc in the arc model $\mathcal{A}_G$, statement (ii) of Observation 4.1 implies that the arcs of $cw_o(x)$ and $ccw_o(x)$ are not covered, which in turn implies that the arcs of $ccw_o(cw_o(x))$ and $cw_o(ccw_o(x))$ are not covered either. ∎

For a circular-arc graph with arc model without a gap, Lemma 4.1 implies that there always exists a minimum paired-dominating set containing a pair of matched vertices x, y forming an overlapping pair such that $x = ccw_o(y)$ and $y = cw_o(x)$; Lemma 4.2 considers such a case. We note that this does not imply that all pairs of matched vertices in a minimum paired-dominating set form overlapping pairs. Indeed, there are cases such that no such a minimum paired-dominating set exists; for example, any minimum paired-dominating set for the arc model shown in Figure 3(a) contains vertices u, v, and w, and a neighbor of exactly one among u, v, and w, which forms a nested pair with (its matched neighbor) u, v, and w, respectively. Additionally, Lemma 4.1 in conjunction with the neighborhood partition given in Section 4.1 may also give the impression that one need consider only minimum paired-dominating sets containing an appropriate vertex v, or $ccw_o(v)$, or $cw_o(v)$, or perhaps $ccw_c(v)$ and $cw_c(v)$. However, this is not true as indicated by the example shown in Figure 3(b): as shown, the minimum paired-dominating set is equal to $\{v_2, u_2\}$; yet, the arcs of z, z' can be appropriately rotated so that the minimum paired-dominating set becomes any of the sets $\{v_i, u_i\}$, $i = 1, 2, \ldots, k$. Therefore, without knowing the position of z, z', we need consider all neighbors of vertex w in order to find a minimum paired-dominating set.

Lemma 4.2 *Let G be a circular-arc graph, whose arc model $\mathcal{A}_G$ does not have a gap, and suppose that the adjacent vertices x, y are matched to each other in a perfect matching M of the subgraph of G induced by a minimum paired-dominating set S of G. Further suppose that $x = ccw_o(y)$ and $y = cw_o(x)$ (that is, x forms a clockwise overlapping pair with y) and neither the arc $A(x)$ of x nor the arc $A(y)$ of y are covered by any arc in $\mathcal{A}_G$.*

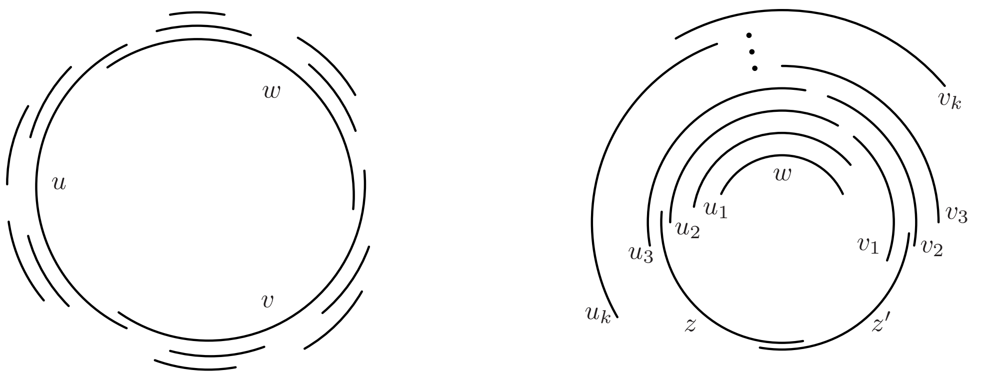

Figure 3: Examples of arc models.

(i) There exists a minimum paired-dominating set $D = D' \cup \{x, y\}$ of G such that

- *D' does not contain any vertices whose arcs in $\mathcal{A}_G$ are covered by the union of the arcs of x and of y;*
- *D' contains at most 1 neighbor of y whose arc extends farther clockwise than the cw_endpoint of the arc of y; symmetrically, D contains at most 1 neighbor of x whose arc extends farther counterclockwise than the ccw_endpoint of the arc of x.*

(ii) Consider the following 4 arc models resulting from $\mathcal{A}_G$:

- *$\mathcal{A}_1$: from $\mathcal{A}_G$ remove the arcs of x, y, and all their neighbors;*
- *$\mathcal{A}_2$: from $\mathcal{A}_G$ remove the arcs of x, y, and all their neighbors except for $ccw_o(x)$;*
- *$\mathcal{A}_3$: from $\mathcal{A}_G$ remove the arcs of x, y, and all their neighbors except for $cw_o(y)$;*
- *$\mathcal{A}_4$: from $\mathcal{A}_G$ remove the arcs of x, y, and all their neighbors except for $cw_o(y)$ and $ccw_o(x)$.*

Then

(a) each of $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, and $\mathcal{A}_4$ has a gap;

(b) there exists a minimum paired-dominating set of G containing the pair $\{x, y\}$ and the smallest among the minimum paired-dominating sets on the graphs corresponding to $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, and $\mathcal{A}_4$ (whenever a paired-dominating set exists).

Proof: Since the arcs of x and y are not covered by any arc in the arc model $\mathcal{A}_G$, statement (ii) of Observation 4.1 implies that the vertices $ccw_o(x)$ and $cw_o(y)$ exist.

(i) Consider the minimum paired-dominating set S and let M be a perfect matching of the induced subgraph $G[S]$. Suppose that S contains a vertex z whose arc is covered by the union of the arcs of x and y in $\mathcal{A}_G$, and let $z' \in S$ be the vertex in S matched to z in M. Clearly, z' must dominate some vertex w not dominated by any other vertex in S, otherwise the set $S - \{z, z'\}$ would also be a paired-dominating set, in contradiction to the minimality of S. The fact that z' dominates w implies that the arc of z' is not covered by the union of the arcs of x and y; additionally, w is not adjacent to either x or y. Then, we can replace z by w in S obtaining a minimum paired-dominating set not containing z (the matched pair $\{z, z'\}$ is replaced by the matched pair $\{z', w\}$). Because we can replace any such vertex z, we can obtain a minimum paired-dominating set S' that does not contain vertices (other than x and y) whose arcs are covered by the union of the arcs of x and y.

Finally, we show the restriction on the number of neighbors of y whose arcs extend farther clockwise than the cw_endpoint of the arc of y; for simplicity, let us call such a neighbor a *cw-neighbor* of y (we note

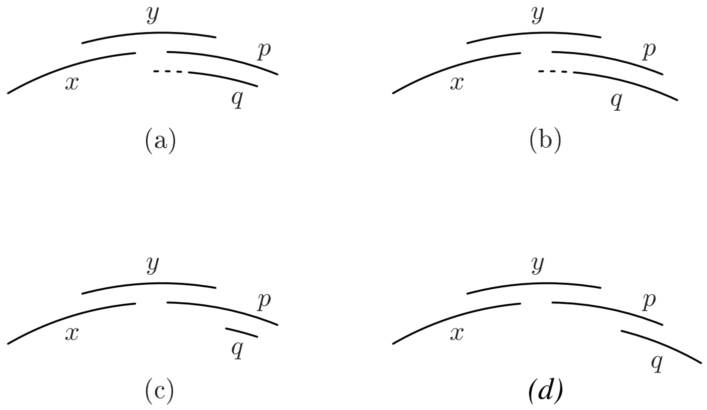

Figure 4: For the proof of Lemma 4.2 (the dashed part of an arc may or may not exist).

that the definition of $y = cw_o(x)$ implies that a cw-neighbor of y is not a neighbor of x). We will show that there exists a minimum paired-dominating set of the graph G which, in addition to not containing vertices whose arcs are covered by the union of the arcs of x and y, contains at most one cw-neighbor of y. Consider the minimum paired-dominating set S' as described in the previous paragraph and let M' be a perfect matching of the induced subgraph $G[S']$. Suppose, for contradiction, that S' contains two cw-neighbors p_1 and p_2 of y. First, suppose that p_1, p_2 are not matched to each other in M' and let $q_1, q_2 \in S'$ be the vertices matched to p_1 and p_2, respectively, in M' (see Figure 4 for the four general cases for the position of each pair p_i, q_i $(i = 1, 2)$ in the arc model $\mathcal{A}_G$ taking into account that $y = cw_o(x)$ and that no vertex in $S' - \{x, y\}$ has its arc covered by the union of the arcs of x and y). The vertices q_1, q_2 are not neighbors of y, and thus the ccw_endpoints of their arcs lie farther clockwise than the cw_endpoint of the arc of y (see Figure 4(c) and (d)). Assume without loss of generality that the cw_endpoint of the union of the arcs of p_2 and q_2 is farther clockwise than the cw_endpoint of the union of the arcs of p_2 and q_2; then, $\big(N[p_1] \cup N[q_1]\big) - \big(N(x) \cup N(y)\big) \subseteq \big(N[p_2] \cup N[q_2]\big) - \big(N(x) \cup N(y)\big)$ which implies that the set $S' - \{p_1, q_1\}$ is a paired-dominating set of G, in contradiction to the minimality of S'. Suppose now that p_1, p_2 are matched to each other in M', and assume without loss of generality that the arc of p_2 extends farther clockwise than the arc of p_1 (see Figure 4(b) for $p = p_1$ and $q = p_2$), which implies that $N[p_1] - N(y) \subseteq N[p_2] - N(y)$. The vertex p_2 dominates a vertex, say, w, not dominated by the elements of $S' - \{p_1, p_2\}$; otherwise, the set $S' - \{p_1, p_2\}$ is a paired-dominating set of G, in contradiction to the minimality of S'. Then, if we replace p_1 by w in S', we obtain a minimum paired-dominating set of G containing only one cw-neighbor of y; note that w is not a neighbor of y since it is not dominated by any element of $S' - \{p_1, p_2\}$.

Therefore, S' contains at most one cw-neighbor of y. A symmetric argument works for the case of neighbors of x whose arcs extend farther counterclockwise than the ccw_endpoint of the arc of x.

(ii) (a) Since $y = cw_o(x)$, the arc of $cw_o(y)$ cannot extend farther counterclockwise than the cw_endpoint of the arc of x; additionally, since $x = ccw_o(y)$, the arc of $ccw_o(x)$ cannot extend farther clockwise than the ccw_endpoint of the arc of y. Then, since the cw_endpoint of the arc of x lies in the arc of y, each of the arc models $\mathcal{A}_i$ $(i = 1, 2, 3, 4)$ has a gap in a clockwise direction from the ccw_endpoint of the arc of y to the cw_endpoint of the arc of x.

(b) Let D be a minimum paired-dominating set of the graph G as described in statement (i) of the lemma. We have the following cases for D.

1. If D contains no neighbor of x other than y and no neighbor of y other than x, then the set $D - \{x, y\}$ is a paired-dominating set of the graph G_1 with arc model $\mathcal{A}_1$; in fact, $D - \{x, y\}$ is a minimum paired-dominating set of G_1 since if there were a smaller paired-dominating set X of G_1, then $X \cup \{x, y\}$ would be a paired-dominating set of G in contradiction to the minimality of D.

2. If D contains one neighbor of y (other than x) whose arc extends farther clockwise than the cw_endpoint of the arc $A(y)$ of y and no neighbor of x other than y, then the set $D - \{x, y\}$ is a minimum paired-dominating set of the graph with arc model $\mathcal{A}_2$.

3. Similarly to the previous case, if D contains one neighbor of x (other than y) whose arc extends farther counterclockwise than the ccw_endpoint of the arc $A(x)$ of x and no neighbor of y other than x, then the set $D - \{x, y\}$ is a minimum paired-dominating set of the graph with arc model $\mathcal{A}_3$.

4. Finally, if D contains one neighbor of x (other than y) whose arc extends farther counterclockwise than the ccw_endpoint of the arc $A(x)$ of x, and one neighbor of y (other than x) whose arc extends farther clockwise than the cw_endpoint of the arc $A(y)$ of y, then the set $D - \{x, y\}$ is a minimum paired-dominating set of the graph with arc model $\mathcal{A}_4$. ∎

4.3 The Algorithm

As mentioned above, the idea behind our algorithm is to reduce the problem to a paired-domination on an interval graph by appropriately creating a gap in the arc model of the input circular-arc graph G. In order to create a gap, we take advantage of the fact that for any vertex $v \in V(G)$, at least one among v and its neighbors belongs to each paired-dominating set. Thus we pick an appropriate[1] vertex v and for each vertex $x \in N[v]$, we apply Lemma 4.1 so that if the arc of x is not covered we consider minimum paired-dominating sets containing either $\{ccw_o(cw_o(x)), cw_o(x)\}$ or $\{ccw_o(x), cw_o(ccw_o(x))\}$, whereas if the arc of x is covered we consider minimum paired-dominating sets containing either $\{ccw_o(cw_o(z)), cw_o(z)\}$ or $\{ccw_o(z), cw_o(ccw_o(z))\}$ where $z = cw_c(x)$. Then, for each such pair, we apply Lemma 4.2 obtaining four arc models with a gap, which can be turned into interval models and the paired domination problem can be solved on each of them in $O(n)$ time using the algorithm of the previous section. A description of the overall algorithm in pseudocode is given below where we also detect the existence of isolated vertices; Procedure SOLUTION_CONTAINING_VERTEX applies Lemmas 4.1 and 4.2.

Algorithm CIRC_ARC_PAIRED DOMINATION($\mathcal{A}_G$)
Input : an arc model $\mathcal{A}_G$ of a circular-arc graph G with arc endpoints sorted
Output : a minimum paired-dominating set of G, if one exists, or
a message that there is no solution and an isolated vertex of G

---

1. {*compute useful information and check for a gap and isolated vertices*}
   check for a gap in the arc model $\mathcal{A}_G$ and find a vertex v of minimum degree in G;
   if the degree of v is 0
   then **print**("No solution...");
   print that v is an isolated vertex and **exit**;
   if $\exists$ a gap in $\mathcal{A}_G$ (say, next to the ccw_endpoint of the arc $A(w)$)
   then $D \leftarrow$ INTERVAL_PAIRED_DOMINATION($\mathcal{I}_G$) where $\mathcal{I}_G$ is an interval model corresponding to the arcs in $\mathcal{A}_G$ starting at the ccw_endpoint of $A(w)$ and moving clockwise;
   go to Step 3;

2. {*v: a vertex of minimum degree in G*}
   find $cw_c(v)$ (if it exists);
   if $cw_c(v)$ does not exist {*arc $A(v)$ not covered in $\mathcal{A}_G$*}
   then $D \leftarrow$ paired-domin. set returned by Procedure MIN_P-D-SET_CONTAINING_VERTEX($\mathcal{A}_G$,v);
   else {*Procedure* MIN_P-D-SET_CONTAINING_VERTEX *on $cw_c(v)$ will be called...*}
   {*...in the for-loop below when $w = cw_c(v)$*}

[1] In order to get a good time complexity, in our algorithm we choose as v the vertex of minimum degree in G.

$D \leftarrow V(G)$;

for each neighbor w of v **do**

find $cw_c(w)$ (if it exists);

if $cw_c(w)$ does not exist {*arc* $A(w)$ *not covered in* $\mathcal{A}_G$}

then $D' \leftarrow$ paired-domin. set returned by MIN_P-D-SET_CONTAINING_VERTEX($\mathcal{A}_G$, w);

else $D' \leftarrow$ paired-domin. set returned by MIN_P-D-SET_CONTAINING_VERTEX($\mathcal{A}_G$, $cw_c(w)$);

$D \leftarrow$ minimum between D and D';

end-for

3. **print**("A minimum paired-dominating set of the input graph is:");
   print the elements of the set D.

Procedure MIN_P-D-SET_CONTAINING_VERTEX($\mathcal{A}_G$,w)

Input : an arc model $\mathcal{A}_G$ of a circular-arc graph G without isolated vertices and a vertex w of G whose arc is not covered by any arc in $\mathcal{A}_G$

Output : a minimum paired-dominating set of G among those containing w

---

1. {*try the overlapping pair* $\{ccw_o(cw_o(w)), cw_o(w)\}$}
   find $cw_o(w)$ and $ccw_o(cw_o(w))$ and assign $y \leftarrow cw_o(w)$ and $x \leftarrow ccw_o(y)$;
   {*x forms a clockwise overlapping pair with y, and $y = cw_o(x)$ and $x = ccw_o(y)$*}
   $\mathcal{A}_1 \leftarrow$ arc model obtained by $\mathcal{A}_G$ after having removed the arcs of x, y, and their neighbors except for $ccw_o(x)$ and $cw_o(y)$;
   $\mathcal{I}_a \leftarrow$ interval model corresponding to the arcs in $\mathcal{A}_1$ starting at the cw_endpoint of the arc $A(x)$ of x and moving clockwise;
   $\mathcal{I}_b \leftarrow$ interval model obtained from $\mathcal{I}_a$ after having removed the interval corresponding to $cw_o(y)$;
   $\mathcal{I}_c \leftarrow$ interval model obtained from $\mathcal{I}_a$ after having removed the interval corresponding to $ccw_o(x)$;
   $\mathcal{I}_d \leftarrow$ interval model obtained from $\mathcal{I}_a$ after having removed the intervals corresponding to $ccw_o(x)$ and $cw_o(y)$;
   $D_1 \leftarrow \{x, y\} \cup$ smallest among the minimum paired-dominating sets (whenever they exist) returned by Algorithm INTERVAL_PAIRED_DOMINATION when applied on $\mathcal{I}_a$, $\mathcal{I}_b$, $\mathcal{I}_c$, and $\mathcal{I}_d$;

2. {*try the overlapping pair* $\{ccw_o(w), cw_o(ccw_o(w))\}$}
   repeat Step 1 for $x \leftarrow ccw_o(w)$ and $y \leftarrow cw_o(y)$ obtaining a paired-dominating set D_2;

3. $D \leftarrow$ minimum between the paired-dominating sets D_1 and D_2;
   report the vertices in D as a minimum paired-dominating set of the graph G that contains x.

We note that the problems on some of the interval models produced may not admit a solution as the removal of the neighbors of x and y may leave isolated vertices; in such a case, another interval model produces the final minimum paired-dominating set.

The correctness of Algorithm CIRC_ARC_PAIRED_DOMINATION follows from Lemmas 4.1 and 4.2.

Time and Space Complexity

Let n and m be the numbers of vertices and edges, respectively, of the given graph G. First, we note that each call to Procedure MIN_P-D-SET_CONTAINING_VERTEX takes $O(n)$ time: Step 1 of the procedure involves identifying $y = cw_o(w)$ and then $x = ccw_o(y)$ (by twice examining all the vertices in the graph in $O(n)$ time), constructing 4 interval models which can be obtained in $O(n)$ time, and applying Algorithm INTERVAL_PAIRED_DOMINATION on each of them, which also takes $O(n)$ time (see Theorem 3.1); similarly, Step 2 also takes $O(n)$ time, as does Step 3. The time complexity of Procedure

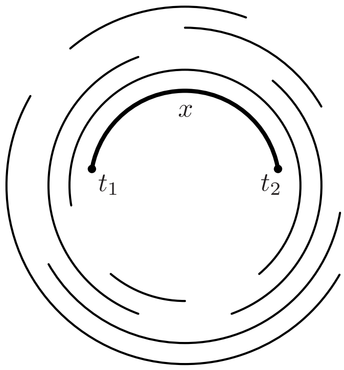

Figure 5: An arc model to illustrate the degree computation.

Min_P-D-Set_containing_Vertex implies that Step 2 of Algorithm Circ_Arc_Paired_Domination takes $O(n+m)$ time: $O(n)$ time is needed for identifying all neighbors of v and for computing $cw_c(v)$ (by comparing the arc of v to all other arcs in the arc model $\mathcal{A}_G$) and $O\big(n\,(1+degree(v))\big)$ time for all the calls to Procedure Min_P-D-Set_containing_Vertex and for all the "minimum between D and D'" computations; because v is a vertex of minimum degree in G, its degree is at most $2m/n$ since $2m = \sum_{x \in V(G)} degree(x) \geq n\, degree(v)$.

Let us now show how to check whether the arc model $\mathcal{A}_G$ has a gap and how to find a vertex of minimum degree in G. The idea is to process all the arc endpoints and maintain the number of *active* arcs, that is, the number of arcs containing the current arc endpoint in their interior (hence we exclude the arc ending at that endpoint). Then, if before processing the ccw_endpoint of an arc, the number of active arcs is 0, we conclude that there exists a gap counterclockwise from that ccw_endpoint. Moreover, the degree of a vertex can be computed by observing that the set of neighbors of a vertex x with arc $A(x)$ is precisely the *disjoint union* of the set of vertices corresponding to the arcs containing the ccw_endpoint of $A(x)$ in their interior and the set of vertices with arcs whose ccw_endpoint belongs to the interior of $A(x)$. The cardinality of the former set of vertices (neighbors of x) is equal to the number of active arcs while processing the ccw_endpoint of $A(x)$; the cardinality of the latter set of vertices can be computed by keeping count of the ccw_endpoints encountered. In particular, if we first encounter the ccw_endpoint t_1 of the arc of x and then its cw_endpoint t_2, then it is not difficult to see that the degree of x is

$$degree(x) \;=\; active\_num\_at\_t_1 + ccw\_num\_at\_t_2 - ccw\_num\_at\_t_1$$

where ccw_num is the number of arc ccw_endpoints encountered and $ccw\_num\_at\_t_1$ is equal to the number of ccw_endpoints encountered including t_1 (since t_2 is a cw_endpoint the number of ccw_endpoints does not change while processing t_2). (For example, in Figure 5, let k be the number of ccw_endpoints encountered when reaching (and including) the ccw_endpoint t_1 of the arc of x ($1 \leq k \leq 6$); then, the number of ccw_endpoints at t_2 is $k+3$ and since the number of arcs containing t_1 in their interior is 3, the degree of x is correctly found equal to $3 + (k+3) - k = 6$.) On the other hand, if we first encounter the cw_endpoint t_2 of the arc $A(x)$ of x and then its ccw_endpoint t_1, then the degree of x is

$$degree(x) \;=\; active\_num\_at\_t_1 + n - (ccw\_num\_at\_t_1 - ccw\_num\_at\_t_2)$$

where $ccw\_num\_at\_t_1$ is again equal to the number of ccw_endpoints encountered including t_1; note that $ccw\_num\_at\_t_1 - ccw\_num\_at\_t_2$ is equal to the number of arcs whose ccw_endpoints do *not* belong to the interior of $A(x)$ and thus by subtracting this number from n gives the number of arcs with their ccw_endpoints in the interior of $A(x)$. (For example, in Figure 5, let k be the number of ccw_endpoints encountered when reaching the cw_endpoint t_2 of the arc of x ($1 \leq k \leq 3$); then, the number of c-cw_endpoints at (and including) t_1 is $k+6$ and since the total number of arcs is 9 and the number of arcs containing t_1 in their interior is 3, the degree of x is correctly found equal to $3 + 9 - ((k+6) - k) = 6$.)

In order to be able to compute the degrees of vertices as presented above (from which we will obtain a vertex of minimum degree):

- we count the number *ccw_num* of ccw_endpoints starting the count at an arbitrary ccw_endpoint;
- we maintain the number *active_num* of active arcs (the value of *active_num* at the first endpoint processed is computed by examining all the arcs of the arc model in order to find those containing that endpoint);
- with each vertex x, we maintain the number x.*endpoints_met* of endpoints of the arc $A(x)$ of x encountered (initialized to 0), the value x.*ccw_num_at_ccw_endp* of *ccw_num* at and including the ccw_endpoint of $A(x)$, and the value x.*active_num_at_prev_endp* of *active_num* at the endpoint of $A(x)$ encountered first.

In detail, the algorithm to check for a gap in the arc model and to find a vertex of minimum degree is as follows:

```
for each vertex w of G do
      w.endpoints_met ← 0;
v ← an arbitrary vertex of G;
active_num ← number of arcs containing the ccw_endpoint of the arc A(v) of v (excluding A(v));
min_degree ← n;
ccw_num ← 0;
for each arc endpoint t starting at the ccw_endpoint of A(v) and moving clockwise do
      w ← vertex of G such that t is an endpoint of A(w) in A_G ;
      w.endpoints_met ← w.endpoints_met + 1;
      if t is the ccw_endpoint of A(w)
      then ccw_num ← ccw_num + 1;
            if active_num = 0
            then there exists a gap next to the ccw_endpoint of arc A(w);
                  exit the for-loop;
            if w.endpoints_met = 2
            then degree ← active_num + n − (ccw_num − w.ccw_num_at_prev_endp);
            else   w.active_num_at_ccw_endp ← active_num;        {first endpoint met}
            active_num ← active_num + 1;        {a new arc has been encountered}
      else {t is the cw_endpoint of arc A(w)}
            if w.endpoints_met = 2
            then degree ← w.active_num_at_ccw_endp + ccw_num − w.ccw_num_at_prev_endp;
            active_num ← active_num − 1;        {an arc has ended}
      if w.endpoints_met = 1        {first endpoint met}
      then w.ccw_num_at_prev_endp ← ccw_num;
      if degree < min_degree        {minimum degree calculation}
      then min_degree ← degree;
            min_degree_v ← w;
end-for
```

The correctness of the above procedure follows from the discussion preceding the pseudocode and the fact that both endpoints of each arc will be processed implying that the degrees of all the vertices will be computed and will be taken into account in the minimum degree computation.

Initializing the values of the fields *endpoints_met* for each vertex and computing the initial value of *active_num* take $O(n)$ total time. Assuming that each arc endpoint is associated with the vertex whose arc ends at that endpoint, then each iteration of the for-loop takes $O(1)$ time. Therefore, the above computation takes a total of $O(n)$ time and so does Step 1 of Algorithm Circ_Arc_Paired_Domination.

Finally, Step 3 takes $O(n)$ time. The space needed by the algorithm is $O(n)$. In total, Algorithm Circ_Arc_Paired_Domination takes $O(n+m)$ time using $O(n)$ space.

Summarizing, we have the following theorem:

Theorem 4.1 *Let G be a circular-arc graph with no isolated vertices. Then, given an arc model of G with the arc endpoints sorted, Algorithm* Circ_Arc_Paired_Domination *computes a minimum-cardinality paired-dominating set of G in $O(n+m)$ time and $O(n)$ space.*

Since an arc model corresponding to a circular-arc graph can be computed from the graph in time linear in its size [27, 22], we conclude that, given a circular-arc graph, we can compute a minimum-cardinality paired-dominating set of the graph in $O(n+m)$ time, where n is the number of vertices and m is the number of edges of the graph.

5 Concluding Remarks

In this paper we studied the paired domination problem on interval and circular-arc graphs and presented $O(n)$ and $O(n+m)$-time algorithms, respectively, given an interval or an arc model representation with endpoints sorted; our results improve on previous $O(n+m)$ and $O(m(n+m))$-time algorithms [7].

An interesting open question is to investigate whether the paired domination problem on circular-arc graphs can be solved in $O(n)$ time. The case of Figure 3(b) seems to imply that a new different approach will be needed to obtain an $O(n)$-time algorithm.

Additionally, it would also be interesting to find optimal or at least better algorithms for the paired domination problem on other classes of graphs.

Acknowledgments The author would like to thank the anonymous referees for bringing [6] to his attention and for their suggestions that improved the presentation and readability of the paper.

References

[1] A.A. Bertossi, Total domination in interval graphs, Inform. Process. Lett. 23, 131–134, 1986.

[2] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, J. Comput, System Sci. 13, 335–379, 1976.

[3] B. Brešar, M.A. Henning, and D.F. Rall, Paired-domination of Cartesian products of graphs and rainbow domination, Electr. Notes in Discrete Math. 22, 233–237, 2005.

[4] M.-S. Chang, Efficient algorithms for the domination problems on interval and circular-arc graphs, SIAM J. Comput. 27(6), 1671–1694, 1998.

[5] L. Chen, C. Lu, and Z. Zeng, A linear-time algorithm for paired-domination problem in strongly chordal graphs, Inform. Process. Lett. 110, 20–23, 2009.

[6] L. Chen, C. Lu, and Z. Zeng, Labeling algorithms for paired-domination problems in block and interval graphs, J. Comb. Optim. 19, 457–470, 2010.

[7] T.C.E. Cheng, L.Y. Kang, and C.T. Ng, Paired domination on interval and circular-arc graphs, Discrete Appl. Math. 155, 2077–2086, 2007.

[8] T.C.E. Cheng, L. Kang, and E. Shan, A polynomial-time algorithm for the paired-domination problem on permutation graphs, Discrete Appl. Math. 157, 262–271, 2009.

[9] P. Dorbec, S. Gravier, and M.A. Henning, Paired-domination in generalized claw-free graphs, J. Comb. Optim. 14, 1–7, 2007.

[10] E.M. Eschen and J.P. Spinrad, An $O(n^2)$ algorithm for circular-arc graph recognition, Proc. Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 128-137, 1993.

[11] M. Farber, Domination, independent domination, and duality in strongly chordal graphs, Discrete Appl. Math. 7, 115–130, 1984.

[12] O. Favaron and M.A. Henning, Paired-domination in claw-free cubic graphs, Graphs and Combinatorics 20, 447–456. 2004.

[13] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, Inc., New York, 1980.

[14] M. Habib, R. McConnell, C. Paul, and L. Viennot, Lex-BFS and partition refinement with applications to transitive orientation, interval graph recognition and consecutive ones testing, Theor. Comput. Sci. 234(1-2), 59–84, 2000.

[15] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.

[16] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater, Domination in Graphs: Advanced Topics, Marcel Dekker, New York, 1998.

[17] T.W. Haynes and P.J. Slater, Paired-domination in graphs, Networks 32, 199–206, 1998.

[18] S.T. Hedetniemi and R. Laskar (eds.), Topics on Domination, Ann. Discrete Math. 48, North-Holland, Amsterdam, 1991.

[19] M.A. Henning and M.D. Plummer, Vertices contained in all or in no minimum paired-dominating set of a tree, J. Comb. Optim. 10, 283–294, 2005.

[20] W.-L. Hsu and K.-H. Tsai, Linear time algorithms on circular-arc graphs, Inform. Process. Lett. 49, 123–129, 1991.

[21] L. Kang, M.Y. Sohn, and T.C.E. Cheng, Paired-domination in inflated graphs, Theor. Comput. Sci. 320, 485–494, 2004.

[22] H. Kaplan and Y. Nussbaum, A simpler linear-time recognition of circular-arc graphs, Proc. Scandinavian Workshop on Algorithm Theory, LNCS 4059, 41–52, 2006.

[23] J.M. Keil, Total domination in interval graphs, Inform. Process. Lett. 22, 171–174, 1986.

[24] N. Korte and R.H. Möhring, An incremental linear-time algorithm for recognizing interval graphs, SIAM J. Comput. 18, 68–81, 1989.

[25] E, Lappas, S.D. Nikolopoulos, and L. Palios, An $O(n)$-time algorithm for the paired domination problem on permutation graphs, Eur. J. Comb. 34(3), 593–608, 2013.

[26] C.L. Lu, M-T. Ko, and C.Y. Tang, Perfect edge domination and efficient edge domination in graphs, Discrete Appl. Math. 119, 227–250, 2002.

[27] R. McConnell, Linear-time recognition of circular-arc graphs, Algorithmica 37, 93–147, 2003.

[28] H. Qiao, L. Kang, M. Gardei, and D.-Z. Du, Paired-domination of trees, J. Global Optimization 25, 43–54, 2003.

[29] G. Ramalingam and C. Pandu Rangan, Total domination in interval graphs revisited, Inform. Process. Lett. 27, 271–274, 1988.

[30] G. Ramalingam and C. Pandu Rangan, A unified approach to domination problems on interval graphs, Inform. Process. Lett. 27, 271–274, 1988.

[31] K. White, M. Farber, and W. Pulleyblank, Steiner trees, connected domination and strongly chordal graphs, Networks 15, 109–124, 1985.

[32] M. Yannakakis and F. Gavril, Edge domination sets in graphs, SIAM J. Appl. Math. 38, 364–372, 1980.