An O(n)-Time Algorithm for the Paired-Domination Problem on Permutation Graphs

Evaggelos Lappas, Stavros D. Nikolopoulos, and Leonidas Palios

Department of Computer Science, University of Ioannina
GR-45110 Ioannina, Greece
{elappas,stavros,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 permutation graphs. We define an embedding of permutation graphs in the plane which enables us to obtain an equivalent version of the problem involving points in the plane, and we describe a sweeping algorithm for this problem; if the permutation over the set $N_n = \{1, 2, \ldots, n\}$ defining a permutation graph G on n vertices is given, our algorithm computes a paired-dominating set of G in $O(n)$ time, and is therefore optimal.

Keywords: permutation graphs, paired-domination, domination, algorithms, complexity.

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 [11] for a long bibliography on domination). The problem finds many applications, most notably in relation to area monitoring problems by the minimum number of guards: the potential guard locations are vertices of a graph in which two locations are adjacent if a guard in one of them monitors the other; then, the minimum dominating set of the graph determines the locations to place the guards.

The domination problem admits many variants; the most basic ones include: domination, edge domination, weighted domination, independent domination,

J. Fiala, J. Kratochvíl, and M. Miller (Eds.): IWOCA 2009, LNCS 5874, pp. 368–379, 2009.

connected domination, total/open domination, locating domination, and paired-domination [11,12,13,14,18,30]. 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 [13]; 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.* [23] 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 [16] 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.* [17] 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 a linear-time algorithm for computing a minimum paired-dominating set of an inflated tree. Bounds for the paired-domination number have been established also for claw-free cubic graphs [9], for Cartesian products of graphs [3], and for generalized claw-free graphs [7]. An $O(n + m)$-time algorithm for computing a paired-dominating set of an interval graph on n vertices and m edges, when an interval model for the graph with endpoints sorted is available has been given by Cheng *et al.* [5]; they also extended their result to circular-arc graphs giving an algorithm running in $O(m(m + n))$ time in this case. Very recently, Cheng *et al.* [6] gave an $O(mn)$-time algorithm for the paired domination problem on permutation graphs.

We too consider the paired domination problem on the class of permutation graphs, a well-known subclass of perfect graphs. Given a permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ over the set $N_n = \{1, 2, \ldots, n\}$, we define the n-vertex graph $G[\pi]$ with vertex set $V(G[\pi]) = N_n$ and edge set $E(G[\pi])$ such that $ij \in E(G[\pi])$ if and only if $(i - j)(\pi_i^{-1} - \pi_j^{-1}) < 0$, for all $i, j \in V(G[\pi])$, where π_i^{-1} is the index of the element i in π. A graph G on n vertices is a *permutation graph* if there exists a permutation π on N_n such that G is isomorphic to $G[\pi]$ (the graph $G[\pi]$ is also known as the inversion graph of G [10]). Therefore, in this paper, we assume that a permutation graph $G[\pi]$ is represented by the corresponding permutation π. A lot of research work has been devoted to the study of permutation graphs, and several algorithms have been proposed for recognizing permutation graphs and for solving combinatorial and optimization problems on them both for sequential computation (see for example [22,26,19,28,21]) as well as for parallel (see [15,20,24]). Moreover, in addition to the above mentioned result of Cheng *et al.* [6] on paired domination, several variants of the domination problem have been considered on permutation graphs; see [8,2,1,29,25,4,27].

In this paper, we study the paired-domination problem on permutation graphs following an approach different from that of Cheng *et al.* [6]. We define an embedding of permutation graphs in the plane and show that every permutation graph G with no isolated vertices admits a minimum-cardinality paired-dominating set of a particular form in the embedding of G. We take advantage of this property to describe an algorithm which "sweeps" the vertices of the embedding from left to right and computes a minimum cardinality paired-dominating set if such a set exists; if the permutation over the set $N_n = \{1, 2, \ldots, n\}$ defining a permutation graph on n vertices is given, our algorithm runs in $O(n)$ time using $O(n)$ space. Since for a permutation graph, a defining permutation can be computed in $O(n + m)$ time [19], our algorithm is optimal.

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.

Let $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ be a permutation over the set $N_n = \{1, 2, \ldots, n\}$. A *subsequence* of π is a sequence $\alpha = (\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_k})$ such that $i_1 < i_2 < \cdots < i_k$. If, in addition, $\pi_{i_1} < \pi_{i_2} < \cdots < \pi_{i_k}$, then we say that α is an *increasing subsequence* of π.

A *left-to-right maximum* of π is an element π_i, $1 \leq i \leq n$, such that $\pi_i > \pi_j$ for all $j < i$. The first element in every permutation is a left-to-right maximum. If the largest element is the first, then it is the only left-to-right maximum; otherwise there are at least two (the first and the largest). The increasing subsequence $\alpha = (\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_k})$ is called a *left-to-right maxima subsequence* if it consists of all the left-to-right maxima of π; clearly, $\pi_{i_1} = \pi_1$. For example, the left-to-right maxima subsequence of the permutation $(4, 2, 6, 1, 9, 3, 7, 5, 11, 12, 8, 10)$ is $(4, 6, 9, 11, 12)$.

The *right-to-left minima subsequence* of π is defined analogously: $\alpha' = (\pi_{j_1}, \pi_{j_2}, \ldots, \pi_{j_{k'}})$ is called a right-to-left minima subsequence if it is an increasing subsequence and consists of all the right-to-left minima of π, where an element π_i, $1 \leq i \leq n$, is a *right-to-left minimum* if $\pi_i < \pi_j$ for all $j > i$. The last element in every permutation is a right-to-left minimum, and thus $\pi_{j_{k'}} = \pi_n$. For the permutation $(4, 2, 6, 1, 9, 3, 7, 5, 11, 12, 8, 10)$, the right-to-left minima subsequence is $(1, 3, 5, 8, 10)$.

We will also be considering points in the plane. For such a point p, we denote by $x(p)$ and $y(p)$ the x- and y-coordinate of p, respectively.

An Embedding of Permutation Graphs. Given a permutation π over the set $N_n = \{1, 2, \ldots, n\}$, we define and use an embedding of the vertices of the permutation graph $G[\pi]$ in the plane based on the mapping:

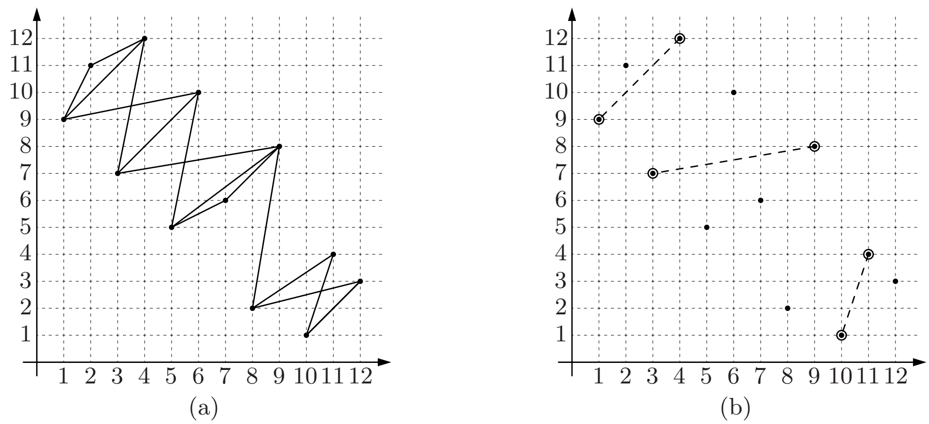

Fig. 1. (a) The embedding of the permutation graph corresponding to the per-mutation $(4, 2, 6, 1, 9, 3, 7, 5, 11, 12, 8, 10)$; (b) A minimum paired-dominating set

$$\text{vertex corresponding to integer } i \quad \longrightarrow \quad \text{point } p_i = (i,\, n + 1 - \pi_i^{-1}). \qquad (1)$$

We note that similar representations have been used by other authors as well; see [1,21]. In our representation, all the points p_i, $1 \leq i \leq n$, are located in the first quadrant of the Cartesian coordinate system and no two such points have the same x- or the same y-coordinate (see Figure 1(a)). Let $P_\pi = \{p_1, p_2, \ldots, p_n\}$. The adjacency condition $ij \in E(G[\pi])$ iff $(i-j)(\pi_i^{-1} - \pi_j^{-1}) < 0$ (for all $i, j \in N_n$) for the permutation graph $G[\pi]$ implies that two points p_i and p_j are adjacent iff $\big(x(p_i) - x(p_j)\big) \cdot \big(y(p_i) - y(p_j)\big) > 0$, i.e., the one of the points is below and to the left of the other. Thus, all the edges have a down-left to up-right direction (Figure 1(a)).

Due to the bijection between the vertices of the permutation graph and the points p_i, with a slight abuse of notation, in the following, we will regard *the points p_i as the vertices of the permutation graph.*

In terms of the above embedding, a point p_i dominates all points $p \in P_\pi$ such that $\big(x(p) - x(p_i)\big) \cdot \big(y(p) - y(p_i)\big) \geq 0$, i.e., p is either below and to the left or above and to the right of p_i (the shaded area in Figure 2 (left)). Then,

Definition 1. *For any edge $e = p_i p_j$, where $p_i, p_j \in P_\pi$, the portion of the plane* covered by *e is the portion of the plane*

$$\{\, q \in \mathbb{R}^2 \mid \big(x(q) - x(p_i)\big) \cdot \big(y(q) - y(p_i)\big) \geq 0 \ \textit{or}\ \big(x(q) - x(p_j)\big) \cdot \big(y(q) - y(p_j)\big) \geq 0 \,\}$$

dominated by p_i or p_j.

The part of the plane not covered by e consists of two disjoint open quadrants, one occupying the upper left corner and the other the bottom right corner.

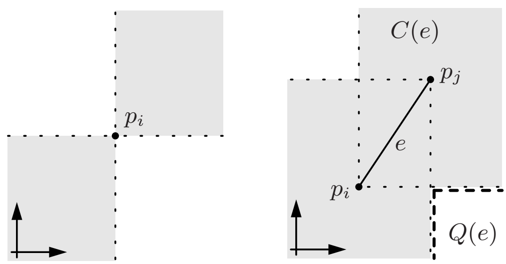

Fig. 2.

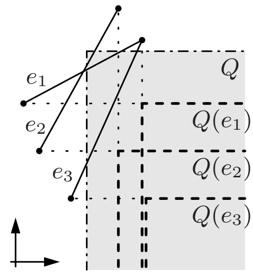

Fig. 3.

To simplify our description, we introduce the following notation (see Figure 2 (right)):

Notation 1. *We denote by*
$C(e)$ *the portion of the plane covered by the edge* e *and*
$Q(e)$ *the bottom right quadrant not covered by* e*.*

Moreover, a left-to-right maximum of a permutation π defining a permutation graph is mapped to a point $p \in P_\pi$ that is a vertex of the upper envelope of the point set P_π $\big($i.e., there does not exist a point $q \in P_\pi - \{p\}$ for which $x(p) \leq x(q)$ and $y(p) \leq y(q)\big)$[1]. For example, the 5 left-to-right maxima of the permutation defining the graph of Figure 1(a) correspond to the points $(4, 12)$, $(6, 10)$, $(9, 8)$, $(11, 4)$, and $(12, 3)$. Similarly, a right-to-left minimum is mapped to a point $p \in P_\pi$ that is a vertex of the lower envelope of the point set P_π (i.e., there does not exist a point $q \in P_\pi - \{p\}$ for which $x(p) \geq x(q)$ and $y(p) \geq y(q)$); the 5 right-to-left minima of the graph of Figure 1(a) correspond to the points $(1, 9)$, $(3, 7)$, $(5, 5)$, $(8, 2)$, and $(10, 1)$ of the lower envelope of P_π. For convenience, *each point in* P_π *corresponding to a left-to-right-maximum (right-to-left minimum, resp.) of a permutation* π *will be called a left-to-right-maximum (right-to-left minimum, resp.) as well.*

Finally, the following result helps us focus on solutions to the paired-domination problem on permutation graphs which are of a particular form, thus enabling us to obtain an efficient algorithm.

Lemma 1. *Let* G *be an embedded permutation graph with no isolated vertices,* $P_\pi = \{p_1, p_2, \ldots, p_n\}$ *the corresponding point set (determined by the mapping in Eq. (1)), and* $u_1, u_2, \ldots, u_\ell$ $(v_1, v_2, \ldots, v_{\ell'}$*, resp.) be the left-to-right maxima (right-to-left minima, resp.) of* P_π *in order from left to right. Then, for any set* A *of edges of* G *whose endpoints dominate the entire point set* P_π*, there exists a matching* M *of edges of* G *such that*

[1] When such inequalities hold for the coordinates of two points p and q, it is often said that q *dominates* p; however, we will avoid using this term so that there is no confusion with the notion of *vertex domination* which is central to our work.

- *the endpoints of the edges in M dominate the entire P_π,*
- $|M| \leq |A|$, *and*
- $M = \{v_{s_1}u_{t_1}, v_{s_2}u_{t_2}, \ldots, v_{s_{|M|}}u_{t_{|M|}}\}$ *where* $s_1 < s_2 < \ldots < s_{|M|} \leq \ell'$ *and* $t_1 < t_2 < \ldots < t_{|M|} \leq \ell$

(i.e., M is a matching which dominates P_π and consists of at most $|A|$ non-crossing *edges each of which connects a left-to-right maximum to a right-to-left minimum of P_π).*

Lemma 1 readily implies the following corollary.

Corollary 1. *Let G be an embedded permutation graph with no isolated vertices, and $P_\pi = \{p_1, p_2, \ldots, p_n\}$ the corresponding point set. Then, G has a paired-dominating set of minimum cardinality whose induced subgraph admits a perfect matching consisting of non-crossing edges of G each of which connects a left-to-right maximum to a right-to-left minimum.*

Such a matching is of the form shown in Figure 1(b). As the edges in such a matching do not cross, they exhibit an ordering from up-left to bottom-right.

3 The Algorithm

Corollary 1 implies that for every permutation graph with no isolated vertices there exists a minimum-cardinality paired-dominating set whose induced embedded subgraph admits a perfect matching of the form shown in Figure 1(b); for a permutation graph G, our algorithm precisely computes a minimum matching M of (the embedded) G of this form whose endpoints dominate all the vertices of G. As the edges in such a matching exhibit an ordering from left to right, our algorithm works by identifying candidates for each edge in M in order from left to right.

In particular, regarding the *leftmost edge* in M, we need to have that

- for each candidate e for the leftmost edge, every point in P_π either is dominated by the endpoints of e or lies in the bottom-right non-covered quadrant $Q(e)$ of e, i.e,

$$P_\pi \ \text{ lies in } \ C(e) \cup Q(e). \tag{2}$$

Furthermore, in order to obtain a minimum-size set M,

- we maintain only the “usefull” partial solutions.

In order to formalize the latter condition, we give the following definition of redundant edges.

Definition 2. *Let G be an embedded permutation graph, Q an open quadrant (bounded only from above and left) which we wish to cover, and $X = \{\, e \in E(G) \mid Q \cap P_\pi = \big(C(e) \cup Q(e)\big) \cap P_\pi \,\}$ (i.e., all the points of P_π belonging to Q lie either in $C(e)$ or in $Q(e)$). Then, we say that an edge $d \in X$ is* redundant *if there exists another edge $d' \in X$ such that $Q(d') \subset Q(d)$.*

For example, in Figure 3, the edges e_1 and e_2 are redundant in light of e_3.

We note that we are interested in minimizing the non-covered part of the plane rather than minimizing the number of points that are not dominated. In light of Definition 2, the fact that we are interested in edges e that minimize the non-covered part $Q(e)$ of the plane is rephrased into that *we are interested in edges e that are not redundant.* The following lemma enables us to identify redundant edges among edges incident on a left-to-right maximum and a right-to-left minimum (see Figure 3):

Lemma 2. *Let G be an embedded permutation graph and let $u_1, u_2, \ldots, u_\ell$ ($v_1, v_2, \ldots, v_{\ell'}$, resp.) be the left-to-right maxima (right-to-left minima, resp.) of G in order from left to right. Moreover, let A be a subset of edges of G which cover the plane except for an open quadrant Q (bounded only from above and left), and $X = \{\, e \in E(G) - A \mid Q \cap P_\pi = \big(C(e) \cup Q(e)\big) \cap P_\pi \,\}$. Then, if X contains an edge $d = v_i u_j$, any edge $v_{i'} u_{j'} \in X - \{d\}$ such that $i' \leq i$ and $j' \leq j$ is redundant.*

Lemma 2 implies that for two edges $v_i u_j$, $v_{i'} u_{j'} \in X$ to be non-redundant, it has to be the case that $(i' - i) \cdot (j' - j) < 0$, that is, the non-redundant edges form a *crossing pattern* like the one shown in Figure 4.

Here is an outline of our algorithm for computing a minimum matching M such that the edges in M are of the form shown in Figure 1(b) and their endpoints dominate all the vertices of the given permutation graph G: The algorithm identifies the non-redundant candidates for the leftmost edge of M and constructs a set $E_1 = \{e_{1,1}, e_{1,2}, \ldots, e_{1,h_1}\}$ of all these candidates. In the general step, we have a set $E_i = \{e_{i,1}, e_{i,2}, \ldots, e_{i,h_i}\}$ of candidates for the i-th edge of the matching M. Then, the algorithm constructs the set E_{i+1} of candidates for the $(i+1)$-st edge by selecting the non-redundant edges among the edges in

$$\{\, e \in \big(E(G) - \textstyle\bigcup_{r=1}^{i} E_r\big) \mid \exists j \text{ such that } Q(e_{i,j}) \cap P_\pi = \big(C(e) \cup Q(e)\big) \cap P_\pi \,\}$$

(i.e., among the edges e such that each of the points that belong to the uncovered quadrant $Q(e_{i,j})$ of an edge $e_{i,j} \in E_i$ is either covered by e or lies in the quadrant $Q(e)$ of e). This is repeated until for some i' and j', the quadrant $Q(e_{i',j'})$ contains no points of P_π. We also ensure that each collected candidate edge $e \in E_{i+1}$ ($i > 1$) has a pointer *back* which points to an edge $e' \in E_i$ such that $Q(e') \cap P_\pi = \big(C(e) \cup Q(e)\big) \cap P_\pi$; then, starting from $e_{i',j'}$ (whose quadrant $Q(e_{i',j'})$ contains no points of P_π), we follow *back*-pointers collecting the edges we visit, thus constructing the matching M that we seek.

The correctness of the algorithm is established by induction on the size of any solution to the paired-domination problem on the input permutation graph G and follows from the correctness of the procedures to compute the set E_1 of candidate edges for the leftmost edge of a solution and to compute the set E_{i+1} of candidates from the corresponding set E_i. We give details on these two procedures in the following paragraphs. For simplicity, we introduce the following additional notation:

Notation 2. *For a point $p \in P_\pi$, we denote by*
$lrmax\_above[p]$ the lowest left-to-right maximum above p and
$rlmin\_left[p]$ the rightmost right-to-left minimum to the left of p.

3.1 Computing the Set E_1

The goal in the construction of the set E_1 is that each edge $e \in E_1$ is incident on a right-to-left-minimum and a left-to-right maximum, is not redundant, and satisfies Eq. (2). Let v_i be a right-to-left minimum. The other endpoint of an edge in E_1 incident on v_i has to be adjacent to v_i and to all the points in P_π to the left of v_i (which are not dominated by v_i); therefore, it needs to be above and to the right of the highest point, say p, among v_i and all the points to the left of v_i. Then, if u_{q_i} is the *lowest* left-to-right maximum above p, each of the left-to-right maxima $u_1, \ldots, u_{q_i}$ will do, whereas none other will do. Yet, among the edges $v_i u_1, \ldots, v_i u_{q_i}$, all but the last one are redundant.

More formally, our observations are summarized in the following lemma:

Lemma 3. *Let G be an embedded permutation graph with no isolated vertices, $P_\pi = \{p_1, p_2, \ldots, p_n\}$ the corresponding point set, and let $u_1, u_2, \ldots, u_\ell$ ($v_1, v_2, \ldots, v_{\ell'}$, resp.) be the left-to-right maxima (right-to-left minima, resp.) in P_π in order from left to right. If $v_r = rlmin\_left[u_1]$, we have:*

- *(i) For each v_i, $i = 1, 2, \ldots, r$, let $p(v_i)$ be the highest among the points in P_π with x-coordinate $\leq x(v_i)$, and let $u_{q_i} = lrmax\_above[p(v_i)]$. Then, for any edge $e_q = v_i u_q$ with $1 \leq q \leq q_i$, it holds that P_π lies in $C(e_q) \cup Q(e_q)$ (i.e., Eq. (2) holds); this does not hold for any edge $e_q = v_i u_q$ with $q > q_i$.*
- *(ii) Among the edges referred to in the statement (i) of the lemma, the edges $v_i u_q$ (where $1 \leq q < q_i$) are all redundant in light of the existence of the edge $v_i u_{q_i}$.*
- *(iii) No edge e incident on a right-to-left minimum to the right of v_r satisfies Eq. (2).*

In Figure 1(a), $v_1 = (1, 9)$, $v_2 = (3, 7)$, and $v_r = v_2$; so, the edges considered are $v_1 u_1, v_1 u_2, v_2 u_1$ (where $u_1 = (4, 12)$ and $u_2 = (6, 10)$), among which $v_1 u_1$ is redundant. We give below the outline of this procedure: in Step 1, we use Lemma 3 to construct a list L of edges satisfying Eq. (2) where L contains exactly the single non-redundant edge incident on each right-to-left minimum to the left of u_1 (see statement (ii) of Lemma 3); in Step 2, we obtain the final set E_1 by removing any redundant edges from L.

Procedure Compute_E_1

1. *highest_p* $\leftarrow p_1$; {*the highest point seen so far is the leftmost point*}
   $L \leftarrow$ a list containing a single node storing the edge connecting p_1 to *lrmax_above*$[p_1]$;
   $i \leftarrow 2$; {*process the points by increasing x-coordinate*}
   while p_i does not coincide with the leftmost left-to-right maximum u_1 **do**
       if $y(p_i) > y(highest\_p)$
       then *highest_p* $\leftarrow p_i$; {*update highest point seen so far*}
       if p_i is a right-to-left minimum
       then insert at the end of L the edge connecting p_i to *lrmax_above* [*highest_p*];
       $i \leftarrow i + 1$;

2. $E_1 \leftarrow \emptyset$;
   let the list L contain the edges $e_1, e_2, \ldots, e_{|L|}$ in order and suppose that $e_i = v_{s_i} u_{t_i}$, where v_{s_i} is a right-to-left minimum and u_{t_i} is a left-to-right maximum;
   $i \leftarrow 1$; {*i indicates position in L of edge checked for inclusion in* E_1}
   while $i < |L|$ **do**
   $j \leftarrow i + 1$;
   {*ignore all edges incident on the same left-to-right maximum...*}
   {*...except for the last one*}
   while $j <= |L|$ **and** $u_{t_j} = u_{t_i}$ **do**
   $j \leftarrow j + 1$;
   add the edge e_{j-1} in E_1 with its *back*-pointer pointing to **NIL**;
   $i \leftarrow j$;
   if $i = |L|$
   then {$i = |L| \Longleftrightarrow e_{|L|-1}$ *is last edge included in* E_1 *and* $u_{t_{|L|-1}} \neq u_{t_{|L|}}$}
   add the edge $e_{|L|}$ in E_1 with its *back*-pointer pointing to **NIL**;

The correctness of Step 1 follows from Lemma 3, statement (ii): for each v_i, we consider only the edge $v_i u_{q_i}$ where $u_{q_i} = lrmax\_above[p(v_i)]$. The correctness of Step 2 follows from Lemma 2; for the correctness of Step 2, it is important to note that because the y-coordinate of point *highest_p* never decreases during the execution of Step 1, the edges $v_{s_i} u_{t_i}$ and $v_{s_j} u_{t_j}$ located in the i-th and j-th node of the list L (for any $i < j$) have $s_i < s_j$ and $t_i \geq t_j$. The edges in the resulting set E_1 form a crossing pattern like the one shown in Figure 4.

3.2 Computing the Set E_{i+1} from E_i

Let $E_i = \{e_{i,1}, e_{i,2}, \ldots, e_{i,h}\}$ be the set of candidate edges for the i-th edge in a minimum matching M such that the edges in M are of the form shown in Figure 1(b) and their endpoints dominate all the vertices of the given permutation graph G. As shown in Figure 4, the quadrants from left to right and from bottom to top are $Q(e_{i,h}), Q(e_{i,h-1}), \ldots, Q(e_{i,1})$, respectively.

For the construction of E_{i+1}, we are interested in non-redundant edges e incident on a right-to-left minimum and on a left-to-right maximum such that there exists $e_{i,j} \in E_i$ for which all the points in $Q(e_{i,j}) \cap P_\pi$ are either covered by e or lie in the bottom right uncovered quadrant $Q(e)$, i.e.,

$$Q(e_{i,j}) \cap P_\pi \;=\; \big(C(e) \cup Q(e)\big) \cap P_\pi. \tag{3}$$

This case is a generalization of the case for E_1; this time, however, we are dealing with a number of quadrants $Q(e_{i,j})$. The following lemma gives a complete coverage of all cases.

Lemma 4. *Let G be an embedded permutation graph with no isolated vertices, $P_\pi = \{p_1, p_2, \ldots, p_n\}$ the corresponding point set, and let $u_1, u_2, \ldots, u_\ell$ ($v_1, v_2, \ldots, v_{\ell'}$, resp.) be the left-to-right maxima (right-to-left minima, resp.) in P_π in order from left to right. Suppose further that the set E_i contains the edges $e_{i,1}$,*

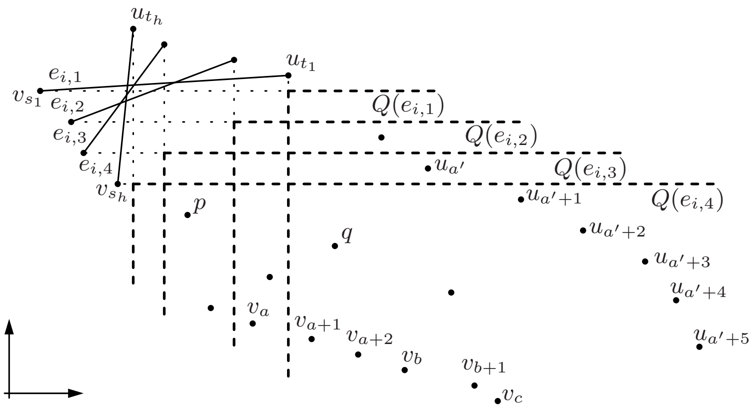

Fig. 4.

$e_{i,2}, \ldots, e_{i,h}$, each incident on a right-to-left minimum and a left-to-right maximum. If $v_a = rlmin\_left[u_{t_1}]$, $u_{a'} = lrmax\_above[v_{s_h}]$, $v_b = rlmin\_left[u_{a'}]$, and $v_c = rlmin\_left[u_{a'+1}]$ (see Figure 4), we have:

(i) The edge connecting v_a to $lrmax\_above[v_a]$ satisfies Eq. (3) for $j = 1$.

(ii) Consider v_k, where $k = a+1, a+2, \ldots, b$. Let $Q(e_{i,r})$ be the rightmost[2] *(i.e., its left side is to the right of the left sides of the other quadrants) among the quadrants that do not contain points $p \in P_\pi$ such that $x(p) < x(v_k)$ and $y(p) > y(v_{s_h})$, and let $u_{q_k} = lrmax\_above[p(v_k)]$ where $p(v_k)$ is the highest point in P_π which belongs to $Q(e_{i,r})$ and is not to the right of v_k. Then, Eq. (3) is satisfied for $Q(e_{i,j}) = Q(e_{i,r})$ and the edge $e = v_k u_{q_k}$; this does not hold for any edge $e = v_k u_q$ with $q > q_k$.*

(iii) Consider v_k, where $k = b+1, b+2, \ldots, c$. Suppose that there exists a quadrant $Q(e_{i,r})$ that contains no points $p \in P_\pi$ such that $y(p) > y(u_{a'+1})$, and let $u_{q_k} = lrmax\_above[p(v_k)]$ where $p(v_k)$ is the highest point in P_π which belongs to $Q(e_{i,r})$ and is not to the right of v_k. Then, Eq. (3) is satisfied for $Q(e_{i,j}) = Q(e_{i,r})$ and the edge $e = v_k u_{q_k}$; this does not hold for any edge $e_q = v_k u_q$ with $q > q_k$.

(iv) Each edge incident on a right-to-left minimum to the left of v_a is redundant. Moreover, for any edge e incident on a right-to-left minimum to the right of v_c, there does not exist $e_{i,j} \in E_i$ that satisfies Eq. (3) with e; in fact, the same holds for any edge e incident on a right-to-left minimum v_k, where $k = b, b+1, \ldots, c$, if every quadrant $Q(\ )$ contains points $p \in P_\pi$ such that $y(p) > y(u_{a'+1})$.

As an example for statement (ii), consider $v_k = v_{a+2}$ in Figure 4: then all 4 quadrants $Q(e_{i,1}), \ldots, Q(e_{i,4})$ contain no points $p \in P_\pi$ such that $x(p) < x(v_k)$

[2] The quadrant $Q(e_{i,r})$ is well defined, since the quadrant $Q(e_{i,h})$ does not contain points $p \in P_\pi$ such that $x(p) < x(v_k)$ and $y(p) > y(v_{s_h})$.

and $y(p) > y(v_{s_h})$; the rightmost quadrant $Q(e_{i,r})$ is $Q(e_{i,1})$, $p(v_k) = q$, and $u_{q_k} = u_{a'+2}$. On the other hand, in the case of $v_k = v_b$, the quadrants $Q(e_{i,1})$ and $Q(e_{i,2})$ contain a point $p \in P_\pi$ such that $x(p) < x(v_k)$ and $y(p) > y(v_{s_h})$; the rightmost quadrant $Q(e_{i,r})$ is $Q(e_{i,3})$, $p(v_k) = p$, and $u_{q_k} = u_{a'+1}$. As an example for statement (iii), we may consider $v_k = v_{b+1}$ or v_c in Figure 4: in either case, $Q(e_{i,r}) = Q(e_{i,4})$, $p(v_k) = p$, and $u_{q_k} = u_{a'+1}$.

Our procedure for computing E_{i+1} takes advantage of Lemma 4. Similarly to Procedure Compute_E$_1$, it works in two steps: in the first step, it constructs a list L containing at most one edge incident on each of the right-to-left minima from v_a (inclusive) to v_b (inclusive), and potentially to v_c (inclusive) depending on whether the conditions of statement (iii) of the lemma hold; next, in a 2nd step, it selects only the non-redundant edges among the edges in L. In more detail, the procedure processes the points in P_π to the right of u_{t_h} up to v_b or v_c from left to right, and maintains in a stack only the quadrants that do not contain any point of P_π above the line $y = y(v_{s_h})$ and stores with each of them its highest point so far. Then, for each right-to-left minimum encountered starting with v_a, it applies statement (i), (ii) or (iii) of Lemma 4.

For the case shown in Figure 4, at the end of the first step, the list L contains the edges $v_a u_{a'+4}$, $v_{a+1} u_{a'+4}$, $v_{a+2} u_{a'+2}$, $v_b u_{a'+1}$, $v_{b+1} u_{a'+1}$, and $v_c u_{a'+1}$. Among them, the edges $v_a u_{a'+4}$, $v_b u_{a'+1}$, and $v_{b+1} u_{a'+1}$ are redundant, so that the final set is $\{v_{a+1} u_{a'+4}, v_{a+2} u_{a'+2}, v_c u_{a'+1}\}$.

3.3 Time and Space Complexity of the Algorithm

Regarding the complexity of our algorithm, we can show the following theorem:

Theorem 1. *Let G be a permutation graph with no isolated vertices determined by a permutation π over the set N_n. Then, given π, our algorithm computes a minimum-cardinality paired-dominating set of G in $O(n)$ time using $O(n)$ space.*

References

1. Atallah, M.J., Manacher, G.K., Urrutia, J.: Finding a minimum independent dominating set in a permutation graph. Discrete Appl. Math. 21, 177–183 (1988)
2. Brandstadt, A., Kratsch, D.: On domination problems for permutation and other graphs. Theoret. Comput. Sci. 54, 181–198 (1987)
3. Brešar, B., Henning, M.A., Rall, D.F.: Paired-domination of Cartesian products of graphs and rainbow domination. Electr. Notes in Discrete Math. 22, 233–237 (2005)
4. Chao, H.S., Hsu, F.R., Lee, R.C.T.: An optimal algorithm for finding the minimum cardinality dominating set on permutation graphs. Discrete Appl. Math. 102, 159–173 (2000)
5. Cheng, T.C.E., Kang, L.Y., Ng, C.T.: Paired domination on interval and circular-arc graphs. Discrete Appl. Math. 155, 2077–2086 (2007)
6. Cheng, T.C.E., Kang, L., Shan, E.: A polynomial-time algorithm for the paired-domination problem on permutation graphs. Discrete Appl. Math. 157, 262–271 (2009)

7. Dorbec, P., Gravier, S., Henning, M.A.: Paired-domination in generalized claw-free graphs. J. Comb. Optim. 14, 1–7 (2007)
8. Farber, M., Keil, J.M.: Domination in permutation graphs. J. Algorithms 6, 309–321 (1985)
9. Favaron, O., Henning, M.A.: Paired-domination in claw-free cubic graphs. Graphs and Combinatorics 20, 447–456 (2004)
10. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, Inc., New York (1980)
11. Haynes, T.W., Hedetniemi, S.T., Slater, P.J.: Fundamentals of Domination in Graphs. Marcel Dekker, New York (1998)
12. Haynes, T.W., Hedetniemi, S.T., Slater, P.J.: Domination in Graphs: Advanced Topics. Marcel Dekker, New York (1998)
13. Haynes, T.W., Slater, P.J.: Paired-domination in graphs. Networks 32, 199–206 (1998)
14. Hedetniemi, S.T., Laskar, R. (eds.): Topics on Domination. Ann. Discrete Math., vol. 48. North-Holland, Amsterdam (1991)
15. Helmbold, D., Mayr, E.W.: Applications of parallel algorithms to families of perfect graphs. Computing 7, 93–107 (1990)
16. Henning, M.A., Plummer, M.D.: Vertices contained in all or in no minimum paired-dominating set of a tree. J. Comb. Optim. 10, 283–294 (2005)
17. Kang, L., Sohn, M.Y., Cheng, T.C.E.: Paired-domination in inflated graphs. Theor. Comput. Sci. 320, 485–494 (2004)
18. Lu, C.L., Ko, M.-T., Tang, C.Y.: Perfect edge domination and efficient edge domination in graphs. Discrete Appl. Math. 119, 227–250 (2002)
19. McConnell, R.M., Spinrad, J.P.: Modular decomposition and transitive orientation. Discrete Math. 201, 189–241 (1999)
20. Nikolopoulos, S.D.: Coloring permutation graphs in parallel. Discrete Appl. Math. 120, 165–195 (2002)
21. Nikolopoulos, S.D., Papadopoulos, C.: On the performance of the First-Fit coloring algorithm on permutation graphs. Inform. Process. Lett. 75, 265–273 (2000)
22. Pnueli, A., Lempel, A., Even, S.: Transitive orientation of graphs and identification of permutation graphs. Canadian J. Math. 23, 160–175 (1971)
23. Qiao, H., Kang, L., Gardei, M., Du, D.-Z.: Paired-domination of trees. J. of Global Optimization 25, 43–54 (2003)
24. Reif, J. (ed.): Synthesis of Parallel Algorithms. Morgan Kaufmann Publishers, Inc., San Mateo (1993)
25. Rhee, C., Liang, Y.D., Dhall, S.K., Lakshmivarahan, S.: An $O(n + m)$-time algorithm for finding a minimum-weight dominating set in a permutation graph. SIAM J. Comput. 25, 404–419 (1996)
26. Spinrad, J.: On comparability and permutation graphs. SIAM J. Comput. 14, 658–670 (1985)
27. Srinivasan, A., Madhukar, K., Nagavamsi, P., Rangan, C.P., Chang, M.-S.: Edge domination on bipartite permutation graphs and cotriangulated graphs. Inform. Process. Lett. 56, 165–171 (1995)
28. Supowit, K.J.: Decomposing a set of points into chains, with applications to permutation and circle graphs. Inform. Process. Lett. 21, 249–252 (1985)
29. Tsai, K.H., Hsu, W.L.: Fast algorithms for the dominating set problem on permutation graphs. Algorithmica 9, 601–614 (1993)
30. Yannakakis, M., Gavril, F.: Edge domination sets in graphs. SIAM J. Appl. Math. 38, 364–372 (1980)