Journal of Informatics and Mathematical Sciences
Vol. 9, No. 3, pp. 635–648, 2017
ISSN 0975-5748 (online); 0974-875X (print)
Published by RGN Publications

http://www.rgnpublications.com

Proceedings of the Conference
Current Scenario in Pure and Applied Mathematics
December 22-23, 2016
Kongunadu Arts and Science College (Autonomous)
Coimbatore, Tamil Nadu, India

Research Article

Roman Domination on Acyclic Permutation Graphs

Angshu Kumar Sinha[1]*, Sachchidanand Mishra[3], Akul Rana[2] and Anita Pal[3]

[1] *Department of Mathematics, NSHM Knowledge Campus, Durgapur 713212, India*
[2] *Department of Mathematics, Narajole Raj College, Paschim Medinipur 721211, India*
[3] *Department of Mathematics, National Institute of Technology, Durgapur 713209, India*
***Corresponding author:** angshusinha20@gmail.com

Abstract. A function $f : V \to [0,1,2]$ is said to be Roman dominating function on a graph $G = (V,E)$ if the function f satisfies the condition that every vertex u for which $f(u) = 0$ has at least one neighboring vertex v with $f(v) = 2$. The weight of a Roman dominating function is the value $f(V) = \sum_{v \in V} f(v)$. The Roman domination number of G is the minimum weight of a Roman dominating function and is denoted by $\gamma_R(G)$. In this paper we study the Roman domination number on acyclic permutation graphs.

Keywords. Design of algorithms; Analysis of algorithms; Permutation graph; Roman domination; Roman domination number

MSC. 05C69; 05C78

Received: January 4, 2017 **Accepted:** Febaruary 28, 2017

1. Introduction

Let $G = (V,E)$ be a simple connected graph with vertex set V and edge set E. A subset D of vertex set V is a dominating set if every vertex $v \in V \setminus D$ is adjacent to at least one vertex of D. The domination number $\gamma(G)$ is the minimum cardinality dominating set of G. A function $f : V \to [0,1,2]$ is defined as a Roman dominating function (*RDF*) on a graph $G = (V,E)$ if the function satisfies the condition that every vertex u for which $f(u) = 0$ has at least one neighboring vertex v with $f(v) = 2$. The weight of a Roman dominating function is the value $f(V) = \sum_{v \in V} f(v)$.

If $f : V \to [0,1,2]$ be a Roman dominating function then V_0, V_1, V_2 be a partition of the vertex set V induced by f, i.e., $f = (V_0, V_1, V_2)$ with $V_i = \{v \in V : f(v) = i\}$ where $i = 0,1,2$.

The set V_2 dominates the set V_0, i.e., every vertex in V_0 is adjacent to a vertex in V_2. The weight of a Roman dominating function is the value $f(V) = \sum_{v \in V} f(v) = 2|V_2| + |V_1|$. The *Roman domination number* (*RDN*) of G is the minimum weight Roman dominating function and is denoted by $\gamma_R(G)$. Cockayne *et al.* observed that $\gamma(G) \le \gamma_R(G) \le 2\gamma(G)$ [3].

A graph G with vertex set $V = \{1,2,3,\ldots,n\}$ is called a permutation graph if there exists a permutation $\pi = \{\pi(1), \pi(2), \ldots, \pi(n)\}$ on V, such that for all $i, j \in V$ and $(i,j) \in E$ if and only if

$$(i-j)(\pi^{-1}(i) - \pi^{-1}(j)) < 0.$$

For each $i \in V$, $\pi^{-1}(i)$ denotes the position of the number i in π. Consider two parallel line segments and consider n points $1,2,3,\ldots,n$ from left to right on each line segment. Let π denote a permutation of $\{1,2,\ldots,n\}$. Draw n line segments, connecting i in top line to point $\pi^{-1}(i)$ in bottom line. Each line segments i represent a vertex and $(i,j) \in E$ if and only if the two line segments $(i, \pi^{-1}(i))$ and $(j, \pi^{-1}(j))$ intersect.

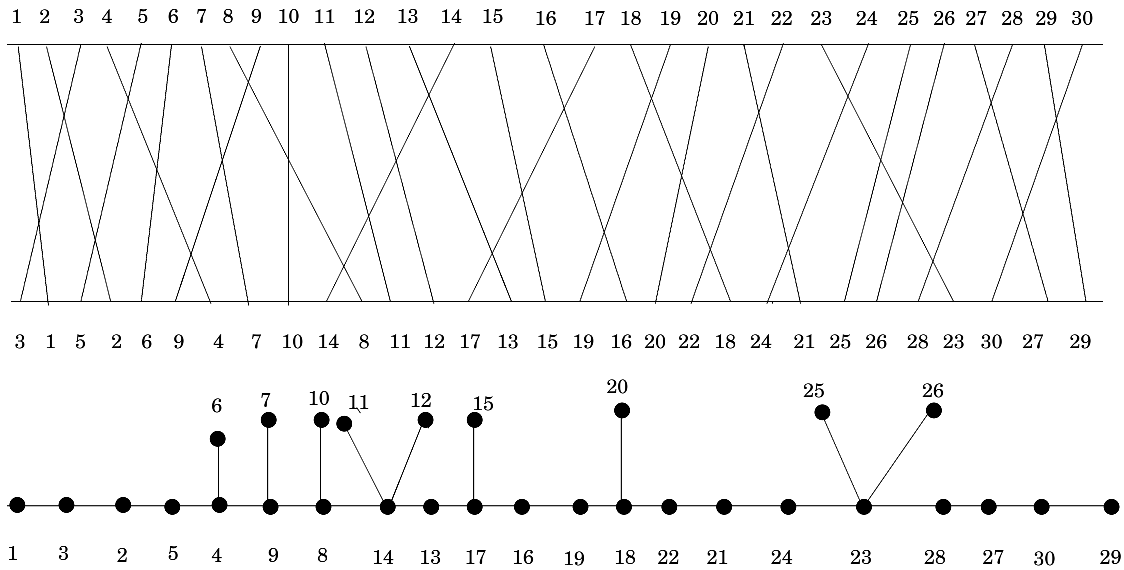

Figure 1. A acyclic permutation graph and its graphical representation

1.1 Review of Previous Work

In 1967 Gallai characterized permutation graphs in terms of forbidden induced sub graphs [7]. Pnnueli *et al.* suggested an algorithm for recognizing permutation graphs, by applying the transitive orientation algorithm to the graph and to its complement [18]. Spinard designed an $O(n^2)$ time algorithm for recognizing permutation graphs [25]. The permutation graph have been widely discussed in the literature, see [1, 8, 14, 15, 17, 19, 20, 22].

Domination and its variations have been extensively studied in the literature, see [2, 10, 11, 23, 24]. Roman domination have both historical and mathematical implication. In 4th century A.D., Constantine the Great (Emperor of Rome) decreed that any city without a legion stationed to secure it must adjacent another city having two stationed legions.

The definition of the *Roman dominating function* (*RDF*) was given implicitly by Stewart [26], Revelle and Rosing [21]. Cockaynea *et al.* [3] studied the graph theoretic properties of this variant of the domination number of a graph. They computed the Roman domination number on path P_n and cycle C_n of any graph G with n vertices. Xing *et al.* [27] gave a characterization of graphs for which $\gamma_R(G)$=$\gamma(G)$+ $K(2 \le K \le \gamma(G))$. The Roman domination function have been widely discussed in the literature, see [4–6, 9, 12, 13, 28].

1.2 Application

Roman domination have military implication to use arm force and ammunition to protect their neighboring area. It may be used to mobilize the rescue operation so that the minimum effort results maximum benefits of the affected people. Permutation graphs have been applied to model and solve problems concerning memory allocation, circuit layout, altitude assignment problem in airlines.

1.3 Main Result

To the best of our knowledge, no algorithm is available to solve Roman domination number on acyclic permutation graph. First we present an $O(n^2)$ time algorithm to construct a tree on acyclic permutation graph. Then we design an $O(n)$ time algorithm for computing the Roman domination numbers on the tree which is obtained from acyclic permutation graph.

1.4 Organization of the Research Work

The remainder of this paper is organized as follows. In Section 2, we introduce the notations and definitions used throughout the paper. In Section 3, we study the approaches towards solving Roman domination number on acyclic permutation graph and present some intermediate results for the same. In Section 4, we present an $O(n^2)$ time algorithm for Roman domination problem on acyclic permutation graph. In Section 5, conclusion is made.

2. Notations and Preliminaries

Let $T(i)$ is the farthest right line segment on the top channel intersecting the line segment i, such that $T(i) > i$ or $T(i) = i$, if such line segment does not exist.

Again $B(i)$ is the farthest left line segment on the bottom channel intersecting the line segment i, such that $B(i) < i$ or $B(i) = i$, if such line segment does not exist.

If $T(i) = j$ and $B(j) = i$, then the vertices from i to j make a span such that span $L(i,j) = \{u : (u,i) \in E$ or $(u,j) \in E$ and $u \leq j\}$, where $i,j = 1,2,3,\ldots,n$, i.e., span $L(i,j)$ is the set of all vertices u such that u is adjacent to i or adjacent to j and $u \leq j$.

Clearly, there is finite number of spans in an acyclic connected permutation graph. Since $T(i) = j$ and $B(j) = i$, are the necessary and sufficient condition to make a span $L(i,j)$, there exist at least two members i and j in every span.

Consider the two successive span $L(i,j+1)$ and span $L(j,k)$ where $T(i) = j+1$ and $B(j+1) = i$; $T(j) = k$ and $B(k) = j$ such that $i < j < j+1 < k$. Here the line segment j and the line segment $j+1$ are the common members of the two successive span $L(i,j+1)$ and span $L(j,k)$. So the span which is lying between two spans have minimum four line segments. But there exists only one span on the right side of the first span and there exists only one span on the left side of the last span. Therefore, in the case of the first span and the last span, there exist minimum three line segments.

If there exist three or four line segments in a span, then the line segments make a path. If a span has more than four line segments then we construct a block B_m such that $B_m =$ span $L(i,j)$. Here B_m denotes the m-th block in G. P_s denotes the path with path length s lie between 1st line segment and 1st block of G. P_b denotes the path lies between two successive block of G where b is the path length. P_t denotes the path with path length t lies between the last block and the last line segment of G.

3. Some Results

Lemma 3.1. *For acyclic permutation graph, if $i < j < k$ and (i,k), $(j,k) \in E$, then $(i,j) \notin E$.*

Proof. If possible, let $(i,j) \in E$. Since (i,k), $(j,k) \in E$ for $i < j < k$, therefore, there exists a clique of order 3. In this case, every vertex is adjacent to each other which makes a circuit.

Therefore in acyclic permutation graph there does not exists any clique of order 3 which makes a circuit. Hence $(i,j) \notin E$. □

In this paper, at first an acyclic permutation graph is converted to a tree. For this, construction of a span is required. There are two type of spans. One makes a path and other makes a block. Properties of span is discussed in the following lemma.

Lemma 3.2. *There exists only one edge between any two successive spans.*

Proof. Let us consider the two successive $span\ L(i, j+1)$ and $span\ L(j,k)$. Here the vertices j and $j+1$ are intersect each other. Hence $(j, j+1) \in E$.

Since the graph is acyclic, there exists one and only one edge between any two vertices. Again, since $(j, j+1) \in E$, there exists only one edge between two successive spans. Hence the lemma.

If the number of vertices in a span is more than four then the vertices make a block B_m. The two successive blocks B_m and B_{m+1} may have two common vertices i and j $(i < j)$ such that they are adjacent to each other.

Definition 3.3. If more than one line segments of the block B_m intersect the first line segment i of the block B_m, then the line segment i is called the core point of the block B_m. Similarly if more than one line segments of the block B_m intersect the last line segment j of the block B_m, then the line segment j is called the core point of the block B_m. It is obvious that in each block there may exist maximum two core points.

We state the following lemma about the core point of a block.

Lemma 3.4. *If j and i are the core point of the blocks B_m and B_{m+1} respectively and if $B_m \cap B_{m+1} = \{i, j\}$, $i < j$, then the core points i and j are adjacent to each other.*

Proof. To prove this lemma, we consider the vertex j as the core point of the block B_m and consider i as the core point of the block B_{m+1}. Since $B_m \cap B_{m+1} = \{i, j\}$ then the core point j of the block B_m is also a member of the block B_{m+1}. Again the vertex i which is the core point of the block B_{m+1} is also a member of the block B_m. This is possible if the vertices i and j are adjacent to each other. Hence the lemma. □

Lemma 3.5. *If j and k are the core point of the blocks B_m and B_{m+1} respectively and if $B_m \cap B_{m+1} = \{i, j\}$ $(i < j < k)$, where $T(i) = k$ then the vertex i is adjacent to both the core points j and k.*

Proof. Consider j and k are the core point of the blocks B_m and B_{m+1} respectively. Since $T(i) = k$, so the vertex i is adjacent to the core point k of the block B_m. Again since $B_m \cap B_{m+1} = \{i, j\}$ and the vertex i is adjacent to the core point k, where $i < j < k$, then the vertex i is adjacent to the core point j. Hence vertex i is adjacent to both the core points j and k. □

There exist three type of paths P_s, P_b and P_t. We state the following lemma about the path P_b which lies between two blocks.

Lemma 3.6. *If the blocks B_m and B_{m+1} has no common vertex, i.e., $B_m \cap B_m = \Phi$, then either there exists a path $P_b (b \geq 2)$ between the two blocks B_m and B_{m+1} or there exists an edge between two blocks.*

Proof. Since the graph is connected, so there must exists at least one vertex i which is adjacent to both the blocks B_m and B_{m+1}. Since i is adjacent to the block B_m, then there exists one edge between i and one of the vertex of block B_m. Again since i is adjacent to the block B_{m+1}, then there exists one edge between i and one of the vertex of block B_{m+1}. Hence the vertex i makes a path P_b of length 2 between the two blocks.

Now if there exist more than one vertex between the two blocks B_m and B_{m+1}, obviously this vertices make a path P_b of length $b > 2$. The path P_b is adjacent to both the blocks. Hence if $B_m \cap B_{m+1} = \Phi$, then there exists a path $P_b(b \geq 2)$.

Now consider the case that there does not exist a path $P_b(b \geq 2)$ between two blocks such that $B_m \cap B_{m+1} = \Phi$. Since the graph is connected, there must exist a vertex $i \in B_m$ and a vertex $j \in B_{m+1}$ such that they are adjacent to each other which make an edge between two blocks. Hence proof the lemma. □

We compute RDN along the paths P_b, P_s, P_t in the following lemma by Induction method. This was stated by Ernie J. Cockaynea *et al.* [3]

Lemma 3.7. *If there exists a path P_b, $b \geq 2$ between the two blocks B_m and B_{m+1} then the value of* RDN *along this path is* $\sum_{v \in P_b} f(v) = \left\lceil \frac{2(b-1)}{3} \right\rceil$.

Proof. Consider a vertex i lies between two blocks B_m and B_{m+1} where $m \geq 1$. Then the vertex i makes a path P_b of length 2 between two blocks B_m and B_{m+1}. By the property of RDF the core point of each block belongs to the set V_2 and adjacent vertices to each core point belong to the set V_0. Therefore obviously i belongs to the set V_1.

Hence $RDN = \sum_{v \in P_{b=2}} f(v) = 1 = \left\lceil \frac{2(2-1)}{3} \right\rceil$.

Again consider the two vertices i and j lie between the two blocks B_m and B_{m+1} such that the two vertices makes a path P_b of length 3. In this case by the definition of RDF, either i or j belongs to V_2.

Hence $RDN = \sum_{v \in P_{b=3}} f(v) = 2 = \left\lceil \frac{2(3-1)}{3} \right\rceil$.

Now if the three vertices i, j, k lie between the two blocks B_m and B_{m+1}, then the three vertices makes a path P_b of length 4. In this case by the same property of RDF, i and k belongs to V_0 and j belongs to V_2.

Hence $RDN = \sum_{v \in P_{b=4}} f(v) = 2 = \left\lceil \frac{2(4-1)}{3} \right\rceil$.

For generalization we can say

(i) if $b = 3p$, then $RDN = \sum_{v \in P_{b=3p}} f(v) = 2p = \left\lceil \frac{2(3p-1)}{3} \right\rceil$, where $p \in z^+$.

(ii) if $b = 3p+1$, then $RDN = \sum_{v \in P_{b=3p+1}} f(v) = 2p = \left\lceil \frac{2((3p+1)-1)}{3} \right\rceil$, where $p \in z^+$.

(iii) if $b = 3p+2$, then $RDN = \sum_{v \in P_{b=3p+2}} f(v) = 2p+1 = \left\lceil \frac{2((3p+2)-1)}{3} \right\rceil$, where $p \in z^+$.

Hence our lemma is proved. □

Lemma 3.8. *If there exists a path P_s between the initial line segment and the first block B_1 of G, then* RDN *along this path is* $\sum_{v \in P_s} f(v) = \left\lceil \frac{2s}{3} \right\rceil$ *here $s \geq 1$.*

Proof. Consider the case that there exists a path P_s between the initial line segment and the first block B_1 of G.

Let i be a vertex adjacent to the block B_1, so i makes a path of length 1 with B_1, then obviously $RDN = \sum_{v \in P_{s=1}} f(v) = 1 = \lceil \frac{2}{3} \rceil$.

Let i, j be the two vertices make a path of length 2 with B_1, then $RDN = \sum_{v \in P_{s=2}} f(v) = 2 = \left\lceil \frac{4}{3} \right\rceil$.

Again let the vertices i, j, k make a path of length 3 with B_1, then $RDN = \sum_{v \in P_{s=3}} f(v) = 2 = \left\lceil \frac{6}{3} \right\rceil$.

For generalization we can say

(i) if $s = 3p$, then $RDN = \sum_{v \in P_{s=3p}} f(v) = 2p = \left\lceil \frac{2.(3p)}{3} \right\rceil$, where $p \in z^+$.

(ii) if $s = 3p+1$, then $RDN = \sum_{v \in P_{s=3p+1}} f(v) = 2p+1 = \left\lceil \frac{2.(3p+1)}{3} \right\rceil$, where $p \in z^+$.

(iii) if $s = 3p+2$, then $RDN = \sum_{v \in P_{s=3p+2}} f(v) = 2p+2 = \left\lceil \frac{2(3p+2)}{3} \right\rceil$, where $p \in z^+$.

Hence our lemma is proved. □

Lemma 3.9. *If there exists a path P_t between the terminal line segment and the last block of G, say B_m, then* RDN *along this path is* $\sum_{v \in P_t} f(v) = \left\lceil \frac{2t}{3} \right\rceil$ *here $t \geq 1$.*

Proof. Proof of this lemma is same as the previous lemma. □

4. Description of Algorithm

4.1 Tree on Acyclic Permutation Graph

At the beginning of our algorithm, we construct the table of $T(i)$ and $B(i)$. From the table, we can easily find the span $L(i,j)$, where $i,j = 1,2,3,\ldots,n$.

If $|\text{span } L(i,j)| \geq 5$ then we construct a block B_m, where $m \geq 1$.

In each block there may exist one or two core points. The core point of the two successive blocks may adjacent to each other or there exists a vertex between two core points of the two successive blocks or there exists a path between two successive blocks.

Again if $3 \leq |\text{span}\, L(i,j)| < 5$, then the line segments of the span make a path. There exist three type of paths:

(i) A path P_s of length s, where $s \geq 1$ may exists between the initial line segment of G and the first block B_1.

(ii) A path P_b of length b, where $b \geq 2$ may exists between any two blocks.

(iii) A path P_t of length t, where $t \geq 1$ may exists between the end block and the terminal line segment G.

Based on the above results and discussion a formal algorithm to construct a tree from the given acyclic permutation graph is presented below.

Algorithm TAPG

Input: A permutation graph $G = (V, E)$ with its permutation representation i, $\Pi(i)$; $i = 1, 2, 3, \ldots, n$.

Output: A tree on acyclic permutation graph.

Step 1: Compute $T(i)$ and $B(i)$, $i = 1, 2, 3, \ldots, n$.

Step 2: If $T(i) = j$ and $B(j) = i$, where $i < j$ and $i, j = 1, 2, 3, \ldots, n$,

then compute span $L(i,j)$.

Step 3: If $|\text{span}\, L(i,j)| \geq 5$.

then go to Step 5.

Step 4: If $3 \leq |\text{span}\, L(i,j)| < 5$,

then go to Step 6.

Step 5: Construct block B_m, $m \geq 1$.

If the first line segment i of B_m intersect with more than one line segment of B_m,

then i is the core point of B_m.

If the last line segment j of B_m intersect with more than one line segment of B_m.

then j is the core point of B_m.

If both the line segments i and j of B_m intersect with

more than one line segments of B_m,

then both the line segments i and j are the core points of B_m.

Step 6: The line segments from i to j of the span $L(i,j)$ make a path.

6.1: If the path exists between the initial line segment of G and the first block B_1, then path is P_s of length s and $s \geq 1$.

If the path exists between the last block B_m and the end line segment of G,
then path is P_t of length t and $t \geq 1$.

6.2: If the path exists between the two blocks,
then path is P_b of length b and $b \geq 2$
else there exists a edge between two blocks.

Step 7: If j and i are the core point of the blocks B_m and B_{m+1} respectively and
if $B_m \cap B_{m+1} = \{i, j\}$ where $i < j$
then i, j are adjacent to each other.

Step 8: If j and k are the core point of the blocks B_m and B_{m+1} respectively and
if $B_m \cap B_{m+1} = \{i, j\}$ where $i < j < k$ and $T(i) = k$
then the vertex i is adjacent to both the core points j and k.

Step 9: If $B_m \cap B_{m+1} = \phi$,
then go to Step 6.2.

Step 10: End *TAPG*.

4.2 Roman Domination Number on Acyclic Permutation Graphs

At first we compute the *RDN* along the paths. If there exists a path P_s of length s, then the value of *RDN* along this path is $\sum_{v \in P_s} f(v) = \left\lceil \frac{2s}{3} \right\rceil$, here $s \geq 1$. Similarly if there exists a path P_t of length t, then the value of *RDN* along this path is $\sum_{v \in P_t} f(v) = \left\lceil \frac{2t}{3} \right\rceil$, here $t \geq 1$. Again if there exists a path P_b of length b between the two consecutive blocks, then the value of *RDN* along this path is $\sum_{v \in P_b} f(v) = \left\lceil \frac{2(b-1)}{3} \right\rceil$, here $b \geq 2$.

Now We assign the value of each core point as 2 and adjacent vertices to the core point assign as 0, i.e., the core point of each block $B_m \in V_2$ and adjacent vertices to the core point $\in V_0$. In this case, the value of the *RDN* of all block is $\sum_{v \in B_m} f(v) = 2|V_2|$, where $m \geq 1$.

Now consider the case when the two successive blocks B_m and B_{m+1} each containing two core points and $B_m \cap B_{m+1} = \{i, j\}$ $(i < j)$, then obviously i, the one of the core points of B_{m+1} and j, one of the core point of B_m are adjacent to each other. In this case by the definition of RDF, if there exist more than one pendent vertices at the vertex i and at the vertex j, then the vertex i and the vertex j are assigned value 2 and their adjacent pendent vertices are assigned value 0. But if i or j or both having only one pendent vertex, then assign value 0 to the core point i or j or both and are assigned value 1 to the pendent vertex at the vertex i or j or both.

In this case, the value of the *RDN* of all block is $\sum_{v \in B_m} f(v) = 2|V_2| + |V_1|$, where $m \geq 1$.

Sum of all *RDN* of blocks is our required *RDN* of the given acyclic permutation graph.

4.3 The Algorithm

A formal description of the algorithm is given below.

Algorithm RDNAPG

Step 1: $RDN = 0$

If there exists a path P_s, $s \geq 1$

then $\sum_{v \in P_s} f(v) = \left\lceil \frac{2s}{3} \right\rceil$

$RDN = RDN + \left\lceil \frac{2s}{3} \right\rceil$.

Step 2: If there exists a path P_t, $s \geq 1$

then $\sum_{v \in P_t} f(v) = \left\lceil \frac{2t}{3} \right\rceil$

$RDN = RDN + \left\lceil \frac{2t}{3} \right\rceil$.

Step 3: If there exists a path P_b, $b \geq 2$

then $\sum_{v \in P_b} f(v) = \left\lceil \frac{2(b-1)}{3} \right\rceil$.

$RDN = RDN + \left\lceil \frac{2(b-1)}{3} \right\rceil$.

Step 4: If $B_k \cap B_{k+1} = \emptyset$

assign 2 to each of the core points of B_m and assign 0 to each neighbor of the core point B_m.

then $\sum_{v \in B_m} f(v) = 2|V_2|$

$RDN = RDN + 2|V_2|$

Step 5: If both the blocks B_m and B_{m+1} each containing two core points

{if $B_m \cap B_{m+1} = \{i, j\}$, i is one of the core point of B_{m+1}

and j is one of the core point of B_m

{if only one pendent vertex is adjacent to i

then assigned 0 to the vertex i and assigned 1 to the pendent vertex of i}

{if only one pendent vertex is adjacent to j

then assigned 0 to the vertex j and assigned 1 to the pendent vertex of j}

then other core points of B_m and B_{m+1} are assigned by 2}

then $RDN = RDN + 2|V_2| + |V_1|$

Step 6: End $RDNAPG$.

4.4 Illustrations of the Algorithm

Values of $T(i)$ and $B(i)$ of the Figure 1 are:

$T(1)=3$, $T(2)=5$, $T(3)=3$, $T(4)=9$, $T(5)=5$, $T(6)=6$, $B(1)=1$, $B(2)=2$, $B(3)=1$, $B(4)=4$, $B(5)=2$, $B(6)=4$, $T(7)=9$, $T(8)=14$, $T(9)=9$, $T(10)=10$, $T(11)=14$, $T(12)=14$, $B(7)=7$, $B(8)=8$, $B(9)=4$, $B(10)=8$, $B(11)=11$, $B(12)=12$, $T(13)=17$, $T(14)=14$, $T(15)=17$, $T(16)=19$, $T(17)=17$, $T(18)=22$, $B(13)=13$, $B(14)=8$, $B(15)=15$, $B(16)=16$, $B(17)=13$, $B(18)=18$, $T(19)=19$, $T(20)=20$, $T(21)=24$, $T(22)=22$, $T(23)=28$, $T(24)=24$, $B(19)=16$, $B(20)=18$, $B(21)=21$, $B(22)=18$, $B(23)=23$, $B(24)=21$, $T(25)=25$, $T(26)=26$, $T(27)=30$, $T(28)=28$, $T(29)=30$, $T(30)=30$, $B(25)=23$, $B(26)=23$, $B(27)=27$, $B(28)=23$, $B(29)=29$, $B(30)=27$.

Now we compute span and block according to our algorithm as follows

span $L(1,3)=\{1,2,3\}$ as $T(1)=3$, $B(3)=1$.

span $L(2,5)=\{2,3,4,5\}$ as $T(2)=5$, $B(5)=2$.

span $L(4,9)=\{4,5,6,7,8,9\}$ as $T(4)=9$, $B(9)=4$.

span $L(8,14)=\{8,9,10,11,12,13,14\}$ as $T(8)=14$, $B(14)=8$.

span $L(13,17)=\{13,14,15,16,17\}$ as $T(13)=17$, $B(17)=13$.

span $L(16,19)=\{16,17,18,19\}$ as $T(16)=19$, $B(19)=16$.

span $L(18,22)=\{18,19,20,21,22\}$ as $T(18)=22$, $B(22)=18$.

span $L(21,24)=\{21,22,23,24\}$ as $T(21)=24$, $B(24)=21$.

span $L(23,28)=\{23,24,25,26,27,28\}$ as $T(23)=28$, $B(28)=23$.

span $L(27,30)=\{27,28,29,30\}$ as $T(27)=30$, $B(30)=27$.

block $B_1=$ span $L(4,9)=\{4,5,6,7,8,9\}$.

Since span $L(4,9)$ has more than 4 line segments, it's make a block B_1.

Again since the line segments 4 and 9 intersect more than one line segments. The line segments 4 and 9 are the core points of block B_1.

In the same way we get the other block.

block $B_2=$ span $L(8,14)=\{8,9,10,11,12,13,14\}$, here the line segments 8 and 14 are the core points of the block B_2.

block $B_3=$ span $L(13,17)=\{13,14,15,16,17\}$, the line segment 17 is the core point of the block B_3

block $B_4=$ span $L(18,22)=\{18,19,20,21,22\}$, here the line segment 18 is the core point of the block B_3.

block $B_5 = \text{span}\, L(23,28) = \{23,24,25,26,27,28\}$, the line segment 23 is the core point of the block B_5.

$B_1 \cap B_2 = 8,9$, the core points 8 and 9 are adjacent to each other.

$B_2 \cap B_3 = 13,14$ and $T(13) = 17$, vertex 13 is adjacent to the core points 14 and 17.

$B_3 \cap B_4 = \phi$, the vertex 16 of the block B_3 and the vertex 19 of the block B_4 are adjacent to each other.

$B_4 \cap B_5 = \phi$, the vertex 21 makes a path P_b of length $b = 2$ between the two blocks B_4, B_5.

The vertices 1, 3 and 2 make a path P_s of length $s = 3$ with the block B_1.

The line segment 27, 30 and 29 make a path P_t of length $t = 3$.

Now we compute the *RDF* on the Figure 1

$$RDN = \sum_{v\in B_m} f(v) = 2|V_2| + |V_1| = 2(5) + 2 = 12,$$

$$RDN = \sum_{v\in P_t} f(v) = \left\lceil \frac{2t}{3} \right\rceil = \left\lceil \frac{2\times 2}{3} \right\rceil = 2,$$

$$RDN = \sum_{v\in P_s} f(v) = \left\lceil \frac{2t}{3} \right\rceil = \left\lceil \frac{2\times 3}{3} \right\rceil = 2,$$

$$RDN = \sum_{v\in P_b} f(v) = \left\lceil \frac{(2b-1)}{3} \right\rceil = \left\lceil \frac{2(2-1)}{3} \right\rceil = 1.$$

Hence the total $RDN = \gamma_R(G) = 12 + 2 + 2 + 1 = 17$.

It is noted that by a theorem of Ore [16], $\gamma(G) \le n/2$ for a connected graph G on n vertices. In our example $n = 30$. So $\gamma(G) \le 15$ and $2\gamma(G) \le 30$. Hence the inequality $\gamma(G) \le \gamma_R(G) = 17 \le 2\gamma(G)$ is verified.

Lemma 4.1. *The RDN on acyclic permutation graph is a minimum.*

Proof. It is obvious that Roman domination number on paths P_s, P_b and P_t are minimum.

Again since the core point of each block $B_m \in V_2$ and consequently all the adjacent vertex to the core point $\in V_0$, therefore *RDN* on each block B_m is minimum. Now consider the two successive blocks B_m and B_{m+1} each containing two core points and $B_m \cap B_{m+1} = \{i, j\}$ $(i < j)$, where i, one of the core points of B_{m+1} and j, one of the core point of B_m. In this case the vertices i and j, and their adjacent vertices are assigned the value $0,1,2$ by the definition of *RDF*. Therefore *RDN* on this two successive blocks is minimum. Hence overall *RDN* on acyclic permutation graph is minimum. □

Theorem 4.2. *Algorithm RDNAPG finds Roman domination number on acyclic permutation graphs in $O(n^2)$ time.*

Proof. The time complexity of algorithm *RDNAPG* is caused mainly by the computation of $T(i)$ and $B(i)$. For each $i \in V$, calculation of $T(i)$, $B(i)$ requires $O(n^2)$ time where n is the total number of line segments. Calculation of span $L(i,j)$ and block B_m takes $O(n)$ time each. Again computation the path P_s, P_t and P_b takes each $O(n)$ time. Hence the overall time complexity to get the acyclic permutation tree is $O(n^2)$. Again computation RDN of acyclic permutation graph takes $O(n)$ time. Thus the overall time complexity is $O(n^2)+O(n)=O(n^2)$. □

5. Conclusion

In this paper, we developed an efficient algorithm that solves the Roman domination number on acyclic permutation graph using $O(n^2)$ time. Future work can be done to investigate the Roman domination number on more general permutation graphs.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] S.C. Barman, S. Mondal and M. Pal, An efficient algorithm to find next-to-shortest path on permutation graphs, Journal of Applied Mathematics and Computing 31 (2009), 369 – 384.

[2] J.G. Chang, Algorithmic aspects of domination in graphs, in *Handbook of Combinatorial Optimization* (D.-Z. Du and P.M. Pardalos eds.), **3**, 339 – 405 (1998).

[3] E.J. Cockayne, P.A. Dreyer Jr., S.M. Hedetniemi and S.T. Hedetniemi, Roman domination in graphs, Discrete Mathematics 278 (2004), 11 – 22.

[4] E.J. Cockayne, P.A. Dreyer, S.M. Hedetniemi and A. McRae, Roman domination in graphs II, Manuscript.

[5] E.J. Cockayne, P.A. Dreyer Jr., S.M. Hedetniemi, S.T. Hedetniemi, and A.A. McRae, The algorithmic complexity of Roman domination, Manuscript.

[6] O. Favaron, H. Karamic, R. Khoeilar and S.M. Sheikholeslami, Note on the Roman domination number of a graph, *Discrete Mathematics* 309 (2009), 3447 – 3451.

[7] T. Gallai, Transitiv orientierbare graphen, *Acta Mathematica Academiae Scientiarum Hungaria* 18 (1967), 25 – 66.

[8] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York (1980).

[9] A. Hansber and L. Volkmann, Upper bounds on the k-domination number and the k-Roman domination number, *Discrete Applied Mathematics* **157** (2009), 1634 – 1639.

[10] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker Inc., New York (1998).

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

[12] M. Liedloffa, T. Kloks, J. Liub and S.L. Peng, Efficient algorithms for Roman domination on some classes of graphs, *Discrete Applied Mathematics* **156** (2008), 3400 – 3415.

[13] C.H. Liu and G.J. Chang, Upper bounds on Roman domination numbers of graphs, *Discrete Mathematics* 312 (2012), 1386 – 1391.

[14] S. Mondal, M. Pal and T.K. Pal, An optimal algorithm for finding depth-first spanning tree on permutation graphs, *Korean Journal of Computational and Applied Mathematics* **6** (1999), 493 – 500.

[15] S. Mondal, M. Pal and T. Pal, Optimal sequential and parallel algorithms to compute a Steiner tree on permutation graphs, *International Journal of Computer Mathematics* **80** (2003), 937 – 943.

[16] O. Ore, Theory of graphs, *Amer. Math. Soc. Colloq. Publ.* **38** (1962), AMS.

[17] M. Pal, A parallel algorithm to generate all maximal independent sets on permutation graphs, *International Journal of Computer Mathematics* **67** (1998), 261 – 274.

[18] A. Pnueli, A. Lempel and S. Even, Transitive Orientation of graphs and identifications of permutation graphs, *Canad. J. Math.* 23 (1971), 160 – 175.

[19] A. Rana, A. Pal and M. Pal, Efficient algorithms to solve k-domination problem on permutation graphs, high performance networking, computing and communication systems, *Communications in Computer and Information Science*, Springer, Berlin — Heidelberg, **163** (2011), 327 – 334.

[20] A. Rana, A. Pal and M. Pal, The 2-neighborhood covering problem on permutation graphs, *Advanced Modeling and Optimization* **13** (2011), 463 – 476.

[21] C.S. ReVelle and K.E. Rosing, Defendens imperium Roman: A classical problem in military strategy, *Amer. Math. Monthly* **107** (2000), 585 – 594.

[22] A. Saha, M. Pal and T.K. Pal, An efficient PRAM algorithm for maximum-weight independent set on permutation graphs, *Journal of Applied Mathematics and Computing* 19 (2005), 77 – 92.

[23] A.K. Sinha, A. Rana and A. Pal, The 2-tuple domination problem on trapezoid graphs, *Annals of Pure and Applied Mathematics* **7** (2014), 71 – 76.

[24] A.K. Sinha, A. Rana and A. Pal, Signed star domination number on proper interval graphs, *International Journal of Pure and Applied Mathematics* **106** (2016), 123 – 129.

[25] J.R. Spinrad, On comparability and permutation graphs, *SIAM Journal of Computing* **14** (1985), 658 – 670.

[26] I. Stewart, Defend the Roman empire!, *Sci. Amer.* **281** (1999), 136 – 139.

[27] H.M. Xing, X. Chen and X.G. Chen, A note on Roman domination in graphs, *Discrete Mathematics* **306** (2006), 3338 – 3340.

[28] F. Xueliang, Y. Yuanshenga and J. Baoqi, Roman domination in regular graphs, *Discrete Mathematics* **309** (2009), 1528 – 1537.