RAIRO-Theor. Inf. Appl.
DOI: 10.1051/ita/2014026

Available online at:
www.rairo-ita.org

DOMINATION PROBLEMS ON $\mathbf{P_5}$-FREE GRAPHS *

Min Chih Lin[1] and Michel J. Mizrahi[1]

Abstract. The minimum roman dominating problem (denoted by $\gamma_R(G)$, the weight of minimum roman dominating function of graph G) is a variant of the very well known minimum dominating set problem (denoted by $\gamma(G)$, the cardinality of minimum dominating set of graph G). Both problems remain NP-Complete when restricted to P_5-free graph class A.A. Bertossi, Dominating sets for split and bipartite graphs. *Inf. Process. Lett.* **19** (1984) 37–40; E.J. Cockayne, P.A. Dreyer Jr., S.M. Hedetniemi and S.T. Hedetniemi, Roman domination in graphs. *Discret. Mathem.* **278** (2004) 11–22.. In this paper we study both problems restricted to some subclasses of P_5-free graphs. We describe robust algorithms that solve both problems restricted to $(P_5, (s,t)$-net)-free graphs in polynomial time. This result generalizes previous works for both problems, and improves existing algorithms when restricted to certain families such as (P_5,bull)-free graphs. It turns out that the same approach also serves to solve problems for general graphs in polynomial time whenever $\gamma(G)$ and $\gamma_R(G)$ are fixed (more efficiently than naive algorithms). Moreover, the algorithms described are extremely simple which makes them useful for practical purposes, and as we show in the last section it allows to simplify algorithms for significant classes such as cographs

Mathematics Subject Classification. — Please, give AMS classification codes —.

Keywords and phrases. Algorithms, dominating set, roman dominating function, P_5-free graphs.

* *Partially supported by UBACyT Grants 20020100100754 and 20020120100058, PICT ANPCyT Grant 1970 and PIP CONICET Grant 11220100100310.*

[1] oscarlin@dc.uba.ar

[2] michel.mizrahi@gmail.com

Article published by EDP Sciences

1. Introduction

The minimum dominating set problem is one of the fundamental problems of graph theory [12] with many applications that arises naturally from many different areas [8, 11, 17]. The problem remains NP-Complete in many restricted graph families such as chordal bipartite graphs [15], planar of maximum degree 3 [10], split and bipartite graphs [2], among others. Note that split graphs is a subclass of P_5-free graphs. On the other hand, polynomial time algorithms has been presented for interval and circular-arc graphs [4], permutation graphs [5], chordal graphs [3], AT-Free graphs [13], (K_p, P_5)-free graphs (for fixed p) [18], and many others.

The minimum roman dominating function problem was introduced as a variant of the minimum dominating set problem. The motivation arises from an optimization problem in location of army to protect the Roman Empire. Each region along with their neighborhood can be protected by one legion (ancient Roman army unit). In case a legion needs to move to a neighbor area, it is required that a second legion remains in the original position to prevent a second attack. The assumption is that two attacks can start at the same time and the army should be prepared to repel them wherever they occur. We refer to [7, 14] for more background on the historical importance and theoretical results for this problem.

We propose very simple non naive algorithms for determining the minimum dominating set and minimum roman dominating function for arbitrary graphs which runs in polynomial time whenever $\gamma(G)$ is a constant. The same algorithms are extended in order to solve these problems efficiently when restricted to P_4-free and $(P_5, (s,t)\text{-net})$-free graphs. We give the definitions of these classes in the next section. Our algorithms improve previous known results for (P_5,bull)-free graphs [13] and (K_p, P_5)-free graphs (for fixed p) [18]. There are already linear-time algorithms to determine $\gamma(G)$ ($\gamma_{_R}(G)$) for any P_4-free G [5,9,16] ([14]). To the best of our knowledge, all of them use some sophisticated structs such as cotrees, modular decompositions, homogeneous extensions, *etc.* or require obtaining an appropriate model from the original graph, and then applying the algorithm. The proposed algorithms are extremely simple and uses the same core procedures, which makes them useful for practical purposes.

2. Preliminaries

Let $G = (V, E)$ be an undirected graph where $V(G)$ and $E(G)$ denote the vertex set and the edge set, respectively. Throughout the paper, $n_G = |V(G)|$ and $m_G = |E(G)|$ denote the numbers of vertices and edges of the graph G. Represent by $N_G(v)$ the subset of vertices adjacent to v, and let $N_G[v] = N_G(v) \cup \{v\}$. Set $N_G(v)$ is called the *neighborhood* of v, while $N_G[v]$ is the *closed neighborhood* of v. Let $S \subseteq V(G)$, denote the *neighborhood* of S as $N_G(S) = \bigcup_{v \in S} N_G(v) \setminus S$ and the *closed neighborhood* of S as $N_G[S] = N_G(S) \cup S$. The *degree* of v is $d_G(v) = |N_G(v)|$. When there is no ambiguity, we may omit the subscripts from n,

m, N and d. Say that u is *universal* when $N[u] = V(G)$. Say that w is *dominated* by vertex v if $N[w] \subseteq N[v]$.

A *dominating set* of G is a set $W \subseteq V(G)$ such that every vertex in $V(G) \setminus W$ is adjacent to some vertex of W. The size of a minimum dominating set in a graph G is called the domination number of G and is denoted as $\gamma(G)$.

As usual, C_n and P_n denote the chordless cycle and the chordless path on n vertices. An induced subgraph H of G is called *dominating* H if $V(H)$ is a dominating set of G. Clearly, if there is a dominating H then $\gamma(G) \leq |V(H)|$.

An (s,t)-net graph is a split graph $G = (K\dot{\cup}I, E)$ where the complete set K is $\{u_1, \ldots, u_s\}$, the independent set I is $\{v_1, \ldots, v_t\}$, $t \leq s$ and $u_iv_j \in E(G)$ iff $i = j$.

A roman dominating function of a graph $G = (V(G), E(G))$ is a function $f : V(G) \rightarrow \{0,1,2\}$ such that every vertex x with $f(x) = 0$ is adjacent to at least one vertex y with $f(y) = 2$. Clearly, considering function f, $V(G)$ can be partitioned into three partitions $V_0 = f^{-1}(0)$, $V_1 = f^{-1}(1)$ and $V_2 = f^{-1}(2)$. Note that the behavior of function f is different from the standard defined functions since it represents a map from numbers to sets of vertices, and f^{-1} returns the mapped vertices to a certain number. The weight of a roman dominating function f is $f(V(G)) = \sum_{x \in V(G)} f(x) = |V_1| + 2|V_2|$. The minimum weight of a roman dominating function of G is called the roman domination number of G and is denoted by $\gamma_{\scriptscriptstyle R}(G)$. It is known that $\gamma(G) \leq \gamma_{\scriptscriptstyle R}(G) \leq 2\gamma(G)$ [14]. Without loss of generality, we assume that G is connected. Therefore, $n \in O(m)$.

We say an algorithm is *robust* if its output is correct even if the input does not belongs to the restricted domain. Thus, whenever this is the case, the algorithm may either (a) correctly solve the problem for the given instance or (b) identify that the input is invalid and report it. There is no guarantee of which of the cases the algorithm will return if an invalid input is given.

3. Algorithms for general graphs

For the algorithms, consider a set $W \subseteq V(G)$. Define $F(W)$ the set of roman domination functions f such that $f^{-1}(2) = W$.

It is easy to see that the function $f \in F(W)$ such that $f^{-1}(0) = N(W)$ and $f^{-1}(1) = V(G) \setminus N[W]$ minimize roman domination weight among functions in $F(W)$. We name this function as f_W. Hence $\gamma_{\scriptscriptstyle R}(G) = \min\limits_{W \subseteq V(G)} f_W(V(G))$.

Let $g : V(G) \rightarrow \{0,1\}$ where $g^{-1}(0) \subseteq N(g^{-1}(1))$ a dominating function where the weight is defined as: $g(V(G)) = \sum\limits_{v \in V(G)} g(v)$

Let $\mathcal{D}$ be the set of all dominating functions and $\mathcal{R}$ the set of all roman dominating functions. Therefore we can conclude:

- $\gamma(G) = \min\limits_{g \in \mathcal{D}} g(V(G))$.
- Given a dominating function g, then $f = 2g$ is a roman dominating function where $f^{-1}(1) = \emptyset$.

- Given a roman dominating function f such that $f^{-1}(1) = \emptyset$ then $g = \frac{f}{2}$ is a dominating function.
- $\gamma(G) = \min_{g \in \mathcal{D}} g(V(G)) = \min_{f \in \mathcal{R} \wedge f^{-1}(1)=\emptyset} \frac{f(V(G))}{2} = \min_{W \subseteq V(G) \wedge f_W^{-1}(1)=\emptyset} \frac{f_W(V(G))}{2}$.

3.1. Domination

We propose a straightforward search algorithm. It consists on looking up each subset of i vertices, which induces a subgraph we name H_i, and check if exists vertex v such that $N[V(H_i) \cup \{v\}] = V(G)$. Note that if such vertex v exists for a given H_i, then $g(V(H_i) \cup \{v\})$ is a dominating function.

We define a procedure called $FindBestAdditionalVertex$ that takes as input a graph G and an induced subgraph of G, named H, and finds a vertex v such that $|N[V(H) \cup \{v\}]|$ is maximized. In order to achieve $O(m)$ time, the procedure should first mark vertices from the set $N[V(H)]$, and then iterate through the neighbors of each candidate vertex (*i.e.* $V(G) \setminus V(H)$) and maintain the vertex with most neighbors outside $N[V(H)]$.

Now we present the algorithm for general graphs. It calls iteratively the described procedure for every induced subgraph of G, named H, in increasing order according to its size. Thus when $V(H)$ is a subset of i vertices the algorithm would discover any dominating set of $i+1$ vertices that includes $V(H)$.

Algorithm 1 GeneralDomination (Graph G)

```
for i ← 0 ... n − 1 do
  for each S_i: subset of i vertices do
    V(H) ← S_i
    v ← FindBestAdditionalVertex(H, G)
    if N[S_i ∪ {v}] = V(G) then
      return S_i ∪ {v}
    end if
  end for
end for
```

Since there are $O(n^i)$ subsets of size at most i, the algorithm running time is $O(n^{\gamma(G)-1}m)$.

3.2. Roman domination

We suppose that $\gamma_R(G) \geq 2$ because $\gamma_R(G) = 1$ iff G is trivial. The problem can be solved using a simple modificated version of $GeneralDomination(G)$. The idea is to use the generated sets S_i of vertices as the set V_2 of the roman dominating function, therefore, exploring every possible roman dominating function by making an exhaustive search for V_2. It is clear that for every S_i of different iteration of $GeneralDomination(G)$, there is a roman dominating function f_{V_2} which is at least as good as any other roman dominating function $f_W, S_i \subset W \wedge |W| = |S_i|+1$

(where v is determined by $FindBestAdditionalVertex$ procedure). We keep the best f_{V_2} as f_Z during whole execution of $GeneralDomination(G)$. The algorithm stops when $|S_i|+1 \geq \frac{f_Z(V(G))}{2}$. Clearly, $f_Z(V(G)) \leq f_W(V(G))$, for any $W \subseteq V(G)$ with $|W| < \frac{f_Z(V(G))}{2}$ because there is some $f_{W'}$ with $f_{W'}(V(G)) \leq f_W(V(G))$ and $|W'| = |W|$ which has been examined before. For any $W \subseteq V(G)$ with $|W| \geq \frac{f_Z(V(G))}{2}$, $f_W(V(G)) \geq 2|W| \geq f_Z(V(G))$. Therefore, f_Z is a minimum roman dominating function and $\gamma_R(G) = f_Z(V(G))$.
The running time is $O(n^{\lfloor \frac{\gamma_R(G)}{2} \rfloor -1}m)$ because the number of S_i's to be considered is at most $O(n^{\lfloor \frac{\gamma_R(G)}{2} \rfloor -1})$ (S_i has at most $\lfloor \frac{\gamma_R(G)}{2} \rfloor - 1$ vertices).

4. Algorithms for $(P_5, (s,t)$-net$)$-free graphs

In this section, we will show algorithms for obtaining $\gamma(G)$ and $\gamma_R(G)$ restricted to $(P_5, (s,t)$-net$)$-free graphs, where $t \leq s$. In case s is fixed, then the algorithm solves the problem in polynomial time.

The arboricity of a graph G is the minimum number of spanning forests needed to cover all the edges of the graph. We use it in order to show an upper-bound of our algorithms. The following lemmas of Chiba and Nishizeki [6] are helpful for our algorithms of this section.

Lemma 4.1. [6] *Given a graph $G = (V(G), E(G))$, $\sum_{uv \in E(G)} \min\{d(u), d(v)\} \leq 2\alpha(G)m$ where arboricity $\alpha(G) \in O(\sqrt{m})$*

Lemma 4.2. [6] *Given a graph $G = (V(G), E(G))$, the number of K_p's of G is $O(\alpha(G)^{\frac{p-2}{2}}m)$ and can be list in $O(p \cdot \alpha(G)^{\frac{p-2}{2}}m)$ time.*

4.1. Domination

Theorem 4.3. [1] *For each graph in the class of P_5-free graphs, there exists a dominating K_p, or a dominating P_3.*

Theorem 4.4. *Let G be a P_5-free graph. If G has not a dominating P_3 and $\gamma(G) \geq 3$ then the following conditions are hold:*

(a) G has a (minimal) dominating complete $K_{p\geq 3}$.
(b) G contains a (p,p)-net as induced subgraphs for all minimal dominating complete K_p of G.

Proof. *(a)* is a direct consequence of Theorem 4.3.

First, we prove that G does not contain a dominating C_4. Suppose there is a dominating $C_4 = (u_1, u_2, u_3, u_4)$. If this dominating set is not minimal, then there is a dominating P_3 or $\gamma(G) \leq 2$, a contradiction. Hence, it must be minimal and there are four vertices: $v_1 \in N(u_1) \setminus (N[u_2] \cup N[u_3] \cup N[u_4])$, $v_2 \in N(u_2) \setminus (N[u_3] \cup N[u_4] \cup N[u_1])$, $v_3 \in N(u_3) \setminus (N[u_4] \cup N[u_1] \cup N[u_2])$ and $v_4 \in N(u_4) \setminus (N[u_1] \cup$

$N[u_2] \cup N[u_3]$. If $v_1v_2 \in E(G)$ then $\{v_1, v_2, u_2, u_3, u_4\}$ induces a P_5, a contradiction. Hence, $v_1v_2 \notin E(G)$. Using similar argument, $v_2v_3 \notin E(G)$. If $v_1v_3 \notin E(G)$ then $\{v_1, u_1, u_2, u_3, v_3\}$ induces a P_5, again this is a contradiction. Thus, $v_1v_3 \in E(G)$. But in this case, $\{v_1, v_3, u_3, u_2, v_2\}$ induces a P_5 which contradicts the P_5-freeness of G. Therefore, C_4 is not a dominating set.

Let $D = \{u_1, \ldots, u_p\}$ be a minimal dominating complete of G. Every vertex $v \notin D$ satisfies: $N(v) \cap D \neq \emptyset$. Since D is minimal, there are at least p vertices $v_1, \ldots, v_p \in V(G) \setminus D$. We assume w.l.o.g. $u_j \in N(v_i)$ iff $i = j$.

Suppose $\exists\ v_iv_j \in E(G)$. Hence $C = \{u_i, u_j, v_j, v_i\}$ induces a C_4. Clearly, C is not a dominating set. There is some vertex $z \in V(G)$ such that $C \cap N(z) = \emptyset$. Thus, $\exists u_k \in D \setminus \{u_i, u_j\}$ such that $z \in N[u_k]$. In this case. $\{z, u_k, u_i, v_i, v_j\}$ induces a P_5, absurd. Therefore $v_iv_j \notin E(G)$ and $\{v_1, \ldots, v_p\}$ is an independent set which means $\{u_1, \ldots, u_p, v_1, \ldots, v_p\}$ induces a (p, p)-net. □

Corollary 4.5. *If G is $(P_5, (s, s)$-net)-free graph then $\gamma(G) \leq \max\{3, s - 1\}$.*

Proof. We suppose that G does not contain a dominating P_3, otherwise $\gamma(G) \leq 3 \leq \max\{3, s - 1\}$. By Theorem 4.4, G has a minimal dominating complete K_p (choose the largest one) and G has a (p, p)-net as induced subgraphs. Clearly, $p \leq s - 1$ and $\gamma(G) \leq p \leq s - 1 \leq \max\{3, s - 1\}$. □

As a consequence of Corollary 4.5, there is a polynomial time algorithm to solve the minimum dominating set problema for $(P_5, (s, t)$-net)-free graphs. Next theorem gives an implementation of such algorithm based on procedure $FindBestAdditionalVertex$ described in Section 3.

Theorem 4.6. *For $(P_5, (s, t)$-net)-free graphs where s is a fixed value and $t \leq s$, the domination problem can be solved in*

- *$O(m)$ time, for $s \leq 2$.*
- *$O(m^2)$ time, for $s \leq 4$.*
- *$O(mn^{s-3} + m^{\frac{s}{2}})$ time, for $s \geq 5$.*

Proof. If $s \leq 2$ then G is P_4-free. There are several linear time algorithm to solve domination problem for P_4-free graphs and we propose a robust linear time algorithm basing on procedure $FindBestAdditionalVertex$ in next section.

For $s \geq 3$, applying procedure $FindBestAdditionalVertex$ with differents induced subgraphs H we can find a minimum dominating set as follows:

(1) Check if $\gamma(G) = 1$. $O(m)$. Using H as the empty set.
(2) Check if $\gamma(G) = 2$. $O(mn)$. Using H as each vertex.
(3) Check if there is a P_3 dominating set (in the positive case, $\gamma(G) = 3$). $O(m^2)$. Using H as any edge.
(4) By Theorem 4.4, there is a (p, p)-net as induced subgraphs of G. Hence, $p \leq s - 1$ and there is dominating K_p which implies that $\gamma(G) \leq s - 1$.
    - If $s \geq 5$, we check if minumum dominating sets have size at most $s - 2$. This can be done in $O(n^{s-3}m)$ using all H with at most $s - 3$ vertices.

- If $\gamma(G) = s-1$ then $p = s-1$. The dominating $K_p = K_{s-1}$ can be found in $O(m^{\frac{s}{2}})$ time using the procedure $FindBestAdditionalVertex$ for each induced K_{s-2} of G. The number of K_{s-2}'s is $O(m^{\frac{s-2}{2}})$ and can be list in $O((s-2)m^{\frac{s-2}{2}})$ time by Lemmas 4.1 and 4.2. □

Since (P_5,bull)-free graphs are $(P_5, (3,2)$-net)-free we can use algorithm from Theorem 4.6.

Corollary 4.7. *Dominating set problem can be solved in $O(m^2)$ in $(P_5, bull)$-free graphs.*

To the best of our knowledge the best algorithm for $(P_5, bull)$-free graphs has $O(n^6)$ time complexity [13].

Lemma 4.8. *All given algorithms are robust.*

Proof. From Theorem 4.6, in case there is not a P_5 nor an (s,t)-net in the graph, algorithms above find a dominating set. Otherwise the input graph has a forbidden structure. Returning a certificate is possible but the complexity of given algorithms may be increased by extra computations to find such certificate. Therefore, returning a certificate is not included in these algorithms. □

4.2. Roman Domination

Using Corollary 4.5 and the fact that $\gamma_R(G) \leq 2\gamma(G)$, we have the following corollary.

Corollary 4.9. *If G is $(P_5, (s,s)$-net)-free graph then $\gamma_R(G) \leq \max\{6, 2s-2\}$.*

Theorem 4.10. *For $(P_5, (s,t)$-net)-free graphs where s is a fixed value and $t \leq s$, the roman domination problem can be solved in*

- *$O(m)$ time, for $s \leq 2$.*
- *$O(mn^2)$ time, for $s \leq 4$.*
- *$O(mn^{s-3} + m^{\frac{s}{2}})$ time, for $s \geq 5$.*

Proof. If $s \leq 2$ then G is P_4-free. In [14], a linear time algorithm basing on cotree is given to solve roman domination problem for P_4-free graphs. Also, we describe a robust linear time algorithm to solve this problem in Section 5.

For $s \geq 3$, applying the following procedure we can find a minimum roman dominating function as follows:

(1) If $\gamma_R(G) \leq 7$, then $\gamma_R(G)$ can be determined in $O(mn^{\lfloor \frac{\gamma_R(G)}{2} \rfloor - 1})$ using the general algorithm described in Subsection 3.2
(2) In this case, $\gamma_R(G) \geq 8$ implying $\gamma(G) \geq 4$. By Theorem 4.4, there is a (p,p)-net as induced subgraphs of G. Hence, $p \leq s-1$ and there is dominating K_p which implies that $\gamma(G) \leq s-1$ and $\gamma_R(G) \leq 2s-2$.

- Check if $\gamma_R(G) \leq 2s-3$. This can be done in $O(n^{s-3}m)$ using the general algorithm.
- If $\gamma_R(G) = 2s-2$ then $\gamma(G) = s-1$ and $p = s-1$. The dominating $K_p = K_{s-1}$ can be found in $O(m^{\frac{s}{2}})$ time as we described in the proof of Theorem 4.6. Clearly, $f_{K_p}(V(G)) = 2s-2$ and f_{K_p} is a minimum roman dominating function.
- If not any minimum dominating set can be found then the input graph has a P_5 or (s,t)-net.

Clearly, the total complexity of the described algorithm is $O(mn^{s-3} + m^{\frac{s}{2}})$. For $s \leq 4$, by Corollary 4.9 $\gamma_R(G) \leq 6$. Therefore, step 2 of above procedure can be omitted and the complexity is reduced to $O(mn^2)$. □

Similarly, the algorithms described in Theorem 4.10 are robust.

Corollary 4.11. *Roman Dominating problem can be solved in $O(mn^2)$ in $(P_5, bull)$-free graphs.*

Again, $(P_5, bull)$-free graphs are $(P_5,(3,2))$-net)-free. We can solve roman domination problem using the $O(mn^2)$ algorithm which is faster than the best known $O(n^6)$ time algorithm [14] for this class of graphs.

5. Algorithms for P_4-free graphs

In this section, we show an extremely simple linear time robust algorithm for both problems restricted to P_4-free graphs using the same approach. Suppose G is connected and $|V(G)| = n > 1$.

(1) If the graph has an universal vertex v then: $\gamma(G) = 1$ ($\{v\}$ is a minimum dominating set) and $\gamma_R(G) = 2$ ($f_{\{v\}}$ is a minimum roman dominating function).
(2) If there is vertex v such that $d(v) = n-2$, and $w \notin N(v)$ then $\gamma(G) = 2$ and $\{v, w\}$ is a minimum dominating set.
The Roman domination number $\gamma_R(G)$ is 3 by defining: $f(v) = 2, f(w) = 1, f(N(v)) = 0$.
(3) Choose an arbitrary vertex v and find $w \in N(v)$ such that $N(v) \cup N(w) = V(G)$. In affirmative case, $\gamma(G) = 2$ with $\{v, w\}$ as a minimum dominating set and $\gamma_R(G) = 4$ with $f_{\{v,w\}}$ as a minimum roman dominating function.
(4) Otherwise, G is not P_4-free. It is clear that if the distance of some pair of vertices $u, z \in V(G)$ is $k \geq 3$ then the shortest path connecting them is an induced P_{k+1}, a contradiction. Hence, every pair of vertices are at distance 1 or 2. The vertices in $U = V(G) \backslash N[v]$ are exactly those vertices at distance 2 from v. Clearly, $U \not\subset N(w)$ of any neighbor w of v, otherwise $N(v) \cup N(w) = V(G)$. Hence, there are $u_1, u_2 \in U$ and $w_1, w_2 \in N(v)$ such that $u_1 \in N(w_1) \setminus N(w_2)$ and $u_2 \in N(w_2) \setminus N(w_1)$. If $w_1 w_2 \in E(G)$ then $\{u_1, w_1, w_2, u_2\}$ induces a P_4, otherwise $\{u_1, w_1, v, w_2\}$ induces a P_4. In any case, we have a contradiction because the existence of an induced P_4. Such induced P_4 can be found in linear time and serves as a negative certificate.

The total running time is $O(m)$. Note that Steps 1 and 3 of this algorithm can be done using two applications of procedure $FindBestAdditionalVertex$ employing $H = \emptyset$ for step 1 and $H = \{v\}$ for Step 3.

References

[1] G. Bacso and Zs. Tuza, Dominating cliques in P_5-free graphs. *Periodica Mathematica Hungarica* **21** (1990) 303–308.

[2] A.A. Bertossi, Dominating sets for split and bipartite graphs. *Inf. Process. Lett.* **19** (1984) 37–40.

[3] K.S. Booth and J. Howard Johnson, Dominating sets in chordal graphs. *SIAM J. Comput.* **11** (1982) 191–199.

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

[5] H.S. Chao, Fang-Rong Hsu and Richard C.T. Lee, An optimal algorithm for finding the minimum cardinality dominating set on permutation graphs. *Discret. Appl. Math.* **102** (2000) 159–173.

[6] N. Chiba and T. Nishizeki, Arboricity and subgraph listing algorithms. *SIAM J. Comput.* **14** (1985) 210–223.

[7] E.J. Cockayne, P.A. Dreyer Jr., Sandra M Hedetniemi and Stephen T Hedetniemi, Roman domination in graphs. *Discret. Mathem.* **278** (2004) 11–22.

[8] C. Cooper, R. Klasing and M. Zito, Lower bounds and algorithms for dominating sets in web graphs. *Internet Mathematics* **2** (2005) 275–300.

[9] M. Farber and J. Mark Keil, Domination in permutation graphs. *J. Algorithms* **6** (1985) 309–321.

[10] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[11] T.W. Haynes, S.M. Hedetniemi, S.T. Hedetniemi and M.A. Henning, Domination in graphs applied to electric power networks. *SIAM J. Discret. Math.* **15** (2002) 519–529.

[12] T.W. Haynes, P.J. Slater and S.T. Hedetniemi, Fundamentals of domination in graphs. 1998. Includes bibliographical references (p. 355-438) and index.

[13] D. Kratsch, Domination and total domination on asteroidal triple-free graphs. *Discret. Appl. Math.* **99** (2000) 111–123.

[14] M. Liedloff, T. Kloks, J. Liu, S.-L. Peng, Efficient algorithms for roman domination on some classes of graphs. *Discret. Appl. Math.* **156** (2008) 3400–3415.

[15] H. Müller and A. Brandstädt, The np-completeness of steiner tree and dominating set for chordal bipartite graphs. *Theor. Comput. Sci.* **53** (1987) 257–265.

[16] F. Nicolai and T. Szymczak, Homogeneous sets and domination: A linear time algorithm for distance – hereditary graphs. *Networks* **37** (2001) 117–128.

[17] J. Wu and H. Li, Domination and its applications in ad hoc wireless networks with unidirectional links. In *International Conference on Parallel Processing, 2000.*

[18] I. E. Zverovich. The domination number of (K_p, P_5)-free graphs. *Australasian J. Combin.* **27** (2003) 95–100.

Received March 28, 2014. Accepted August 29, 2014.