Journal of Algebraic Systems
Vol. 9, No 1, (2021), pp 119-135

ALGORITHMIC ASPECTS OF ROMAN GRAPHS

A. POUREIDI*

ABSTRACT. Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is called a dominating set of G if for every $v \in V \setminus S$ there is at least one vertex $u \in N(v)$ such that $u \in S$. The domination number of G, denoted by $\gamma(G)$, is equal to the minimum cardinality of a dominating set in G. A Roman dominating function (RDF) on G is a function $f : V \to \{0, 1, 2\}$ such that every vertex $v \in V$ with $f(v) = 0$ is adjacent to at least one vertex u with $f(u) = 2$. The weight of f is the sum $f(V) = \sum_{v \in V} f(v)$. The minimum weight of a RDF on G is the Roman domination number of G, denoted by $\gamma_R(G)$. A graph G is a Roman Graph if $\gamma_R(G) = 2\gamma(G)$.

In this paper, we first study the complexity issue of the problem posed in [E. J. Cockayane, P. M. Dreyer Jr., S. M. Hedetniemi and S. T. Hedetniemi, On Roman domination in graphs, *Discrete Math.* 278 (2004), 11–22], and show that the problem of deciding whether a given graph is a Roman graph is NP-hard even when restricted to chordal graphs. Then, we give linear algorithms that compute the domination number and the Roman domination number of a given unicyclic graph. Finally, using these algorithms we give a linear algorithm that decides whether a given unicyclic graph is a Roman graph.

1. Introduction

For notation and terminology not given here we refer to [7]. Let $G = (V, E)$ be a graph with the vertex set V and the edge set E. The *open neighborhood* of a vertex $v \in V$ is $N(v) = \{u \in V : uv \in E\}$. The

---

DOI: 10.22044/jas.2020.8188.1400.
MSC(2010): Primary: 05C85; Secondary: 05C69.
Keywords: Dominating set, Roman dominating function, 3-SAT Problem, unicyclic graph.
Received: 11 March 2019, Accepted: 16 October 2020.
*Corresponding author.

degree of v is $\deg(v) = |N(v)|$. A vertex of degree one is referred as a *leaf*. A path (respectively, cycle) graph of order n is denoted by P_n (respectively, C_n). A graph is unicyclic if it is connected and contains precisely one cycle.

For a graph $G = (V, E)$, a set $S \subseteq V$ is called a *dominating set* (DS) of G if every $v \in V \setminus S$ is adjacent to at least one vertex $u \in S$. Furthermore, if S induces a connected subgraph of G, then S is a *connected dominating set* (CDS) of G. The *domination number* (respectively, *connected domination number*) of G, denoted by $\gamma(G)$ (respectively, $\gamma_c(G)$), is the minimum cardinality of a dominating set (respectively, connected dominating set) of G. A DS of G of minimum cardinality is referred as a $\gamma(G)$-set. A connected DS of G of minimum cardinality is referred as a $\gamma_c(G)$-set.

A function $f : V \to \{0, 1, 2\}$ is a Roman dominating function (RDF) of G if every vertex u with $f(u) = 0$ is adjacent to at least one vertex v with $f(v) = 2$. The *weight* of a RDF f, denoted by $w(f)$, is the sum $f(V) = \sum_{v \in V} f(v)$. The mathematical concept of Roman domination, defined and discussed by Stewart [11], and ReVelle and Rosing [10], and subsequently developed by Cockayne et al. [4]. A hundred papers published on various aspects of Roman domination in graphs, for example [1, 2, 3, 5, 12, 13]. A $\gamma_R(G)$-function is a RDF f on G with $w(f) = \gamma_R(G)$. For a RDF f on G, we denote by V_i (or V_i^f to refer to f) the set of all the vertices of G with label i under f. Thus, a RDF f can be represented by (V_0, V_1, V_2), and we can use the notation $f = (V_0, V_1, V_2)$. A graph G with $\gamma_R(G) = 2\gamma(G)$ is called a *Roman graph*. Cockayne et al. [4] posed the following problem.

Problem 1. Characterize the Roman graphs.

Henning [8] gave a constructive characterization of Roman trees. Liedloff et. al. [9] gave algorithms for computing the Roman domination number of interval graphs and cographs. Also, they gave a linear-time algorithm for recognizing Roman cographs.

In this paper, in Section 3 we prove that the decision problem related to Problem 1 is NP-hard even when restricted to chordal graphs. In Section 4, we give linear algorithms that compute the domination number and Roman domination number of a given unicyclic graph. Finally, using these algorithms we give a linear algorithm that decides whether a given unicyclic graph is a Roman graph.

2. Preliminary

Consider the following family of graphs related to Problem 1:

- Family $\mathcal{F}_{R2}$: the family of all graphs G with $\gamma_R(G) = 2\gamma(G)$.
- Family $\mathcal{F}_{R22c}$: the family of all graphs G with $\gamma_R(G) = 2\gamma(G) = 2\gamma_c(G)$.

Note that $\mathcal{F}_{R22c}$ is an infinite family even when restricted to chordal graphs, since for any positive integer n, if T_n is the tree obtained from P_n by adding three new leaves to any vertex of P_n, then it can be seen that $T_n \in \mathcal{F}_{R22c}$. Also, there are chordal graphs that do not belong to $\mathcal{F}_{R22c}$. It is clear that $\gamma(P_n) \neq \gamma_c(P_n)$ and so $P_n \notin \mathcal{F}_{R22c}$. The following is obvious.

Corollary 2.1. $\mathcal{F}_{R22c} \subseteq \mathcal{F}_{R2}$.

Thus, to prove the NP-hardness of problem of whether a given graph belongs to $\mathcal{F}_{R2}$ we only need to prove the NP-hardness of problem of whether a given graph belongs to $\mathcal{F}_{R22c}$. To this end, we introduce a reduction from 3-SAT Problem. Recall that 3-SAT is the problem of deciding whether a given Boolean formula in 3-conjunctive normal form is satisfiable. It is well-know that 3-SAT Problem is NP-complete [6]. Let $\Phi = \{\mathcal{C}, \mathcal{X}\}$ be an instance in 3-SAT Problem, that is, Φ be Boolean formula in 3-conjunctive normal form. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_l\}$ be a set of $l \geq 1$ clauses over a set $\mathcal{X} = \{x_1, \ldots, x_k\}$ of $k \geq 3$ variables. For each $1 \leq j \leq l$, the clause c_j (consisting of exactly three literals) is of the form $c_j = \{y_{1j}, y_{2j}, y_{3j}\}$, where each of y_{1j}, y_{2j} and y_{3j} is either a variable or the negative of a variable in $\mathcal{X}$.

3. NP-hardness results

Consider the following decision problems.

Roman Graph (RG) Problem:
Instance: A graph G.
Question: Is $G \in \mathcal{F}_{R2}$?

Roman 2Connected-Domination (R2CD) Problem:
Instance: A graph G.
Question: Is $G \in \mathcal{F}_{R22c}$?

Let $\Phi = \{\mathcal{C}, \mathcal{X}\}$ be an instance in 3-SAT Problem. We construct graph G_Φ corresponding to Φ as follows. For each variable x_i, where $1 \leq i \leq k$, we construct a graph G_i as a variable gadget, where G_i is obtained from a path graph of order 2 with vertices u_i^1, u_i^2 such that each

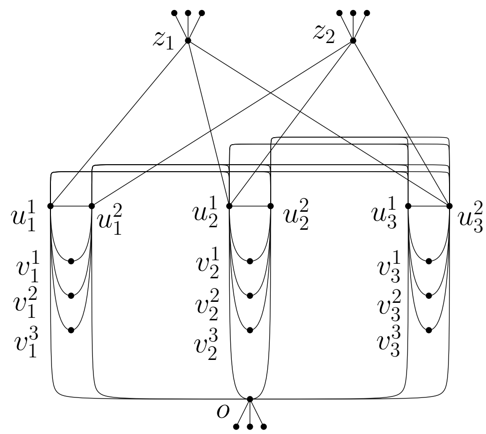

FIGURE 1. Illustrating G_Φ corresponding to $\Phi = \{\{c_1, c_2\}, \{x_1, x_2, x_3\}\}$, where $c_1 = \{\neg x_1, \neg x_2, x_3\}$ and $c_2 = \{x_1, \neg x_2, x_3\}$.

of vertices u_i^1, u_i^2 is adjacent to a new vertex v_i^s for each $s \in \{1, 2, 3\}$. For each clause $c_j = \{y_{1j}, y_{2j}, y_{3j}\}$, where $1 \leq j \leq l$, we add a new vertex z_j such that z_j is adjacent to three new leaves. For $s = 1, 2, 3$, if $y_{sj} = x_i$, for some $1 \leq i \leq k$, then we add an edge $u_i^2 z_j$ and if $y_{sj} = \neg x_i$, for some $1 \leq i \leq k$, then we add an edge $u_i^1 z_j$. We add a new vertex o such that is adjacent to three new leaves and add edges ou_i^1 and ou_i^2 for each $1 \leq i \leq k$. Finally add all edges ab for each $a \in \{u_i^1, u_i^2\}$ and $b \in \{u_j^1, u_j^2\}$ and for all $1 \leq i < j \leq k$. Let G_Φ be the resulting graph. See Figure 1. It is easy to see that G_Φ is a chordal graph.

Lemma 3.1. $\gamma(G_\Phi) = k + l + 1$.

Proof. Let S be a $\gamma(G_\Phi)$-set. Since each of vertices o and z_j, where $1 \leq j \leq l$, is adjacent to three leaves, both $o, z_j \in S$. Since all vertices v_i^1, v_i^2, v_i^3, where $1 \leq i \leq k$, are only adjacent to vertices u_i^1, u_i^2, at least one of vertices u_i^1, u_i^2 belongs to S. So, $\gamma(G_\Phi) = |S| \geq k + l + 1$.

Let $S = \{o, z_j, u_i^1 | 1 \leq i \leq k, 1 \leq j \leq l\}$. Clearly, S is a DS on G_Φ with $|S| = k + l + 1$. So, $\gamma(G_\Phi) \leq k + l + 1$. This completes the proof. $\square$

Lemma 3.2. $\gamma_R(G_\Phi) = 2(k + l + 1)$.

Proof. Let f be a $\gamma_R(G_\Phi)$-function. Since each of vertices o and z_j, where $1 \leq j \leq l$, is adjacent to three leaves, we have $f(o) = f(z_j) = 2$. Since all vertices v_i^1, v_i^2, v_i^3, where $1 \leq i \leq k$, are only adjacent to vertices u_i^1, u_i^2, we find that $\sum_{s=1}^{2} f(u_i^s) + \sum_{s=1}^{3} f(v_i^s) \geq 2$. So, $\gamma_R(G_\Phi) = w(f) \geq 2(k + l + 1)$.

Let $V_2 = \{o, z_j, u_i^1 | 1 \leq i \leq k, 1 \leq j \leq l\}$. Clearly, $f = (V(G_\Phi) - V_2, \emptyset, V_2)$ is a RDF on G_Φ with $w(f) = 2(k + l + 1)$. So, $\gamma_R(G_\Phi) \leq 2(k + l + 1)$. This completes the proof. □

Lemma 3.3. *The Boolean formula Φ is satisfiable if and only if $G_\Phi \in \mathcal{F}_{R22c}$.*

Proof. Assume that Φ is satisfiable. Let T be an assignment of truth values for the variables of $\mathcal{X}$ for which Φ evaluates to *true*. We construct a set S on the vertex set of G_Φ as follows. Initialize S to be $\{o, z_j : 1 \leq j \leq l\}$. If T assigns the value *true* (respectively, the value *false*) to x_i, then we add the vertex u_i^2 (respectively, the vertex u_i^1) to S. It is easy to see that S is a connected DS on G_Φ with $|S| = k + l + 1$. So, $\gamma_c(G_\Phi) \leq k + l + 1$. By Lemma 3.1 we have $\gamma(G_\Phi) = k + l + 1$. By the fact $\gamma(G) \leq \gamma_c(G)$ for any graph G, it obtains that $\gamma_c(G_\Phi) = k + l + 1$. By Lemma 3.2 we have $\gamma_R(G_\Phi) = 2(k+l+1)$. So, $\gamma_R(G_\Phi) = 2\gamma_c(G_\Phi) = 2\gamma(G_\Phi)$, that is, $G_\Phi \in \mathcal{F}_{R22c}$.

Let $G_\Phi \in \mathcal{F}_{R22c}$. By Lemma 3.1 we have $\gamma(G_\Phi) = k + l + 1$. Let S be a connected DS on G_Φ. So, $|S| = k + l + 1$. Clearly, both o and z_j, where $1 \leq j \leq l$, belong to S. Since S is a connected dominating set and o belongs to S, at least one of vertices u_i^1 and u_i^2 belongs to S for each $1 \leq i \leq k$. If both $u_i^1, u_i^2 \in S$ for some $1 \leq i \leq k$, then $|S| > k + l + 1$, a contradiction. So, either both $u_i^1 \in S$ and $u_i^2 \notin S$ or both $u_i^1 \notin S$ and $u_i^2 \in S$ for each $1 \leq i \leq k$.

We fix indices i and j, where $1 \leq i \leq k$ and $1 \leq j \leq l$. Recall that either both $u_i^1 \in S$ and $u_i^2 \notin S$ or both $u_i^1 \notin S$ and $u_i^2 \in S$. If $u_i^1 \notin S$ and $u_i^2 \in S$ (respectively, $u_i^1 \in S$ and $u_i^2 \notin S$), then we assign the value $true$ (respectively, the value $false$) to the variable x_i. We claim that Φ is satisfiable for this assignment.

Assume without loss of generality that $c_j = \{x_1, \neg x_2, x_6\}$. Since $z_j \in S$, we have $u_1^2 \in S$, $u_2^1 \in S$ or $u_6^2 \in S$. Assume without loss of generality that $u_1^2 \in S$. So, x_1 has the value *true*. It causes to satisfy the clause c_j, that is, the Boolean formula Φ is satisfiable. This completes the proof. □

We can compute G_Φ in polynomial time. By Lemma 3.3 and the fact that G_Φ is a chordal graph we have the following result.

Theorem 3.4. *R2CD Problem is NP-hard even when restricted to chordal graphs.*

By Corollary 2.1 and Theorem 3.4 we have the following.

Corollary 3.5. *RG Problem is NP-hard even when restricted to chordal graphs.*

4. Computing Roman domination number of unicyclic graphs

In this section, we give a linear algorithm that computes the Roman domination number of unicyclic graphs. Recall that a connected unicyclic graph is a connected graph with an unique cycle. Let $G = (V, E)$ be a graph with $u \in V$ and let $a \in \{0, 1, 2\}$. We define the following.

- $\gamma_R(G, u = a) = \min\{w(f) | f \text{ is a RDF on } G \text{ with } f(u) = a\}$.

A $\gamma_R(G, u = a)$-function is a RDF f on G with $w(f) = \gamma_R(G, u = a)$ and $f(u) = a$.

Lemma 4.1. *Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \emptyset$ such that $u \in V_1$, $v \in V_2$ and a vertex $w \notin V_1 \cup V_2$. Let $G = (V_1 \cup V_2, E_1 \cup E_2 \cup \{uv\})$. Then, we have the following.*

(*i*) $\gamma_R(G, u = 0) = \min\{\gamma_R(H_1, u = 0) + \gamma_R(H_2, v = 0), \gamma_R(H_1, u = 0) + \gamma_R(H_2, v = 1), \gamma_R(H_1 - u) + \gamma_R(H_2, v = 2)\}$,

(*ii*) $\gamma_R(G, u = 1) = \min\{\gamma_R(H_1, u = 1) + \gamma_R(H_2, v = 0), \gamma_R(H_1, u = 1) + \gamma_R(H_2, v = 1), \gamma_R(H_1, u = 1) + \gamma_R(H_2, v = 2)\}$,

(*iii*) $\gamma_R(G, u = 2) = \min\{\gamma_R(H_1, u = 2) + \gamma_R(H_2 - v), \gamma_R(H_1, u = 2) + \gamma_R(H_2, v = 1), \gamma_R(H_1, u = 2) + \gamma_R(H_2, v = 2)\}$,

(*iv*) $\gamma_R(G - u) = \gamma_R(H_1 - u) + \min\{\gamma_R(H_2, v = 0), \gamma_R(H_2, v = 1), \gamma_R(H_2, v = 2)\}$.

Proof. Let f be a $\gamma_R(G)$-function. Clearly, $f(u) = a$, where $a \in \{0, 1, 2\}$ if and only if both $f(u) = a$ and $f(v) = 0$, both $f(u) = a$ and $f(v) = 1$ or both $f(u) = a$ and $f(v) = 2$. Let f_1, f_2, f_1^u and f_2^v be restrictions of f to H_1, H_2, $H_1 - u$ and $H_2 - v$, respectively. Let g_1^a, g_2^a, g_1^u and g_2^v be a $\gamma_R(H_1, u = a)$-function, $\gamma_R(H_2, v = a)$-function, $\gamma_R(H_1 - u)$-function and $\gamma_R(H_2 - v)$-function, respectively, where $a \in \{0, 1, 2\}$ and let $g_u(u) = g_v(v) = 0$.

Let $f(u) = 0$ and $\gamma_R = \min\{\gamma_R(H_1, u = 0) + \gamma_R(H_2, v = 0), \gamma_R(H_1, u = 0) + \gamma_R(H_2, v = 1), \gamma_R(H_1 - u) + \gamma_R(H_2, v = 2)\}$. So, f_1 is a RDF on H_1 with $f_1(u) = 0$ and f_2 is a RDF on H_2 with $f_2(v) = 0$, function f_1 is a RDF on H_1 with $f_1(u) = 0$ and f_2 is a RDF on H_2 with $f_2(v) = 1$, or f_1^u is a RDF on $H_1 - u$ and f_2 is a RDF on H_2 with $f_2(v) = 2$. Hence, $\gamma_R \leq \gamma_R(G, u = 0)$. Function $g_1 = g_1^0 \cup g_2^0$ is a RDF on G with $g_1(u) = 0$, function $g_2 = g_1^0 \cup g_2^1$ is a RDF on G with $g_2(u) = 0$ and $g_3 = g_1^u \cup g_2^2 \cup g_u$ is a RDF on G with $g_3(u) = 0$. Hence, $\gamma_R(G, u = 0) \leq \gamma_R$. This completes the proof of part (*i*).

Let $f(u) = 1$ and $\gamma_R = \min\{\gamma_R(H_1, u = 1) + \gamma_R(H_2, v = 0), \gamma_R(H_1, u = 1) + \gamma_R(H_2, v = 1), \gamma_R(H_1, u = 1) + \gamma_R(H_2, v = 2)\}$. So, f_1 is a RDF on H_1 with $f_1(u) = 1$ and f_2 is a RDF on H_2 with $f_2(v) = 0$, function f_1 is a RDF on H_1 with $f_1(u) = 1$ and f_2 is a RDF on H_2 with $f_2(v) = 1$ or f_1 is a RDF on H_1 with $f_1(u) = 1$ and f_2 is a RDF on H_2 with $f_2(v) = 2$. Hence, $\gamma_R \leq \gamma_R(G, u = 1)$. Function $g_1 = g_1^1 \cup g_2^0$ is a RDF on G with $g_1(u) = 1$, function $g_2 = g_1^1 \cup g_2^1$ is a RDF on G with $g_2(u) = 1$ and $g_3 = g_1^1 \cup g_2^2$ is a RDF on G with $g_3(u) = 1$. Hence, $\gamma_R(G, u = 1) \leq \gamma_R$. This completes the proof of part (ii).

Let $f(u) = 2$ and $\gamma_R = \min\{\gamma_R(H_1, u = 2) + \gamma_R(H_2 - v), \gamma_R(H_1, u = 2) + \gamma_R(H_2, v = 1), \gamma_R(H_1, u = 2) + \gamma_R(H_2, v = 2)\}$. So, f_1 is a RDF on H_1 with $f_1(u) = 2$ and f_2^v is a RDF on $H_2 - v$, function f_1 is a RDF on H_1 with $f_1(u) = 2$ and f_2 is a RDF on H_2 with $f_2(v) = 1$ or f_1 is a RDF on H_1 with $f_1(u) = 2$ and f_2 is a RDF on H_2 with $f_2(v) = 2$. Hence, $\gamma_R \leq \gamma_R(G, u = 2)$. Function $g_1 = g_1^2 \cup g_2^v \cup g_v$ is a RDF on G with $g_1(u) = 2$, function $g_2 = g_1^2 \cup g_2^1$ a RDF on G with $g_2(u) = 2$ and $g_3 = g_1^2 \cup g_2^2$ is a RDF on G with $g_3(u) = 2$. Hence, $\gamma_R(G, u = 2) \leq \gamma_R$. This completes the proof of part (iii).

Since $G - u = (H_1 - u) \cup H_2$ and graphs $H_1 - u$ and H_2 are disjoint, $\gamma_R(G - u) = \gamma_R(H_1 - u) + \gamma_R(H_2) = \gamma_R(H_1 - u) + \min\{\gamma_R(H_2, v = 0), \gamma_R(H_2, v = 2), \gamma_R(H_2, v = 3)\}$. This completes the proof of part (iv). □

We say that a rooted tree T with the vertex set $V = \{v_1, v_2, \ldots, v_n\}$ has the Property 1, if $j < i$, where $v_j \in V$ is the parent of $v_i \in V$.

Lemma 4.2. *Let T be a tree with $u \in V$. Algorithm 4.1 computes values $\gamma_R(T - u)$ and $\gamma_R(T, u = a)$ for each $a \in \{0, 1, 2\}$ in linear time.*

Proof. We can compute a rooted tree T_u with the root u and Property 1 for T in linear time. Clearly, $\gamma_R(T - u) = \gamma_R(T_u - u)$ and $\gamma_R(T, u = a) = \gamma_R(T_u, u = a)$ for each $a \in \{0, 1, 2\}$. By Lemma 4.1, Algorithm $\mathbf{RD}(T_u)$ returns $(\gamma_R(T_u, u = 0), \gamma_R(T_u, u = 1), \gamma_R(T_u, u = 2), \gamma_R(T_u - u))$. The running time of each iteration of **for** loops of Algorithm $\mathbf{RD}(T_u)$ is $\mathcal{O}(1)$, that is, the running time of Algorithm 4.1 is linear. □

Let $a, b \in \{0, 1, 2\}$, let $G = (V, E)$ be a graph with $u, v \in V$ and a vertex $w \notin V$. We define the following.

- $\gamma_R(G, u = a, v = b) = \min\{w(f) | f$ is a RDF on G with $f(u) = a$ and $f(v) = b\}$,
- $\gamma_R(G, u, w, v = a) = \min\{w(f) | f$ is a RDF on $G + uw$ with $f(u) = 0$, $f(w) = 2$ and $f(v) = b\}$.

Algorithm 4.1: RD(T)

Input: A connected rooted tree $T = (V, E)$ with $V = \{v_1, \ldots, v_n\}$, Property 1 and a vertex $w \notin V$.

Output: $(\gamma_R(T, v_1 = 0), \gamma_R(T, v_1 = 1), \gamma_R(T, v_1 = 2), \gamma_R(T - v_1))$.

for $i = 1$ **to** n **do**
- $\gamma_R(v_i = 0) = \infty$;
- $\gamma_R(v_i = 1) = 1$;
- $\gamma_R(v_i = 2) = 2$;
- $\gamma_R(v_i) = 0$;

for $i = n$ **to** 2 **do**
- Let v_j be the parent of v_i;
- $\gamma_R(v_j = 0) = \min\{\gamma_R(v_j = 0) + \gamma_R(v_i = 0), \gamma_R(v_j = 0) + \gamma_R(v_i = 1), \gamma_R(v_j) + \gamma_R(v_i = 2)\}$;
- $\gamma_R(v_j = 1) = \gamma_R(v_j = 1) + \min\{\gamma_R(v_i = 0), \gamma_R(v_i = 1), \gamma_R(v_i = 2)\}$;
- $\gamma_R(v_j = 2) = \gamma_R(v_j = 2) + \min\{\gamma_R(v_i), \gamma_R(v_i = 1), \gamma_R(v_i = 2)\}$;
- $\gamma_R(v_j) = \gamma_R(v_j) + \min\{\gamma_R(v_i = 0) + \gamma_R(v_i = 1) + \gamma_R(v_i = 2)\}$;

return $(\gamma_R(v_1 = 0), \gamma_R(v_1 = 1), \gamma_R(v_1 = 2), \gamma_R(v_1))$;

Let U be a connected unicyclic graph with the unique cycle $C = v_0, \ldots, v_{k-1}, v_0$, where $k \geq 3$. Let $T(v_0, R) = U - v_0v_1$. Clearly, $T(v_0, R)$ is a tree with the vertex set $V(U)$.

Lemma 4.3. *Let U be a connected unicyclic graph with the unique cycle $v_0, \ldots, v_{k-1}, v_0$ $(k > 2)$. Then, $\gamma_R(U) = \min\{\gamma_R(T(v_0, R), v_0 = a, v_1 = b), \gamma_R(T(v_0, R) - v_0, v_1 = 2), \gamma_R(T(v_0, R) - v_1, v_0 = 2)\}$, where $(a, b) \in \{0, 1, 2\} \times \{0, 1, 2\} - \{(0, 2), (2, 0)\}$.*

Proof. Let $(a, b) \in \{0, 1, 2\} \times \{0, 1, 2\} \setminus \{(0, 2), (2, 0)\}$. Assume that $\gamma = \min\{\gamma_R(T(v_0, R), v_0 = a, v_1 = b), \gamma_R(T(v_0, R) - v_0, v_1 = 2), \gamma_R(T(v_0, R) - v_1, v_0 = 2)\}$.

Let f be a RDF on $T(v_0, R)$ with $w(f) = \gamma_R(T(v_0, R), v_0 = a, v_1 = b)$ and $(f(v_0), f(v_1)) = (a, b)$. Function f is a RDF on U and so $\gamma_R(U) \leq \gamma_R(T(v_0, R), v_0 = a, v_1 = b)$, where $(a, b) \in \{0, 1, 2\} \times \{0, 1, 2\} \setminus \{(0, 2), (2, 0)\}$.

Let f be a RDF on $T(v_0, R) - v_0$ with $f(v_1) = 2$ and $w(f) = \gamma_R(T(v_0, R) - v_0, v_1 = 2)$ and let $g(v_0) = 0$. Function $f \cup g$ is a RDF on U and so $\gamma_R(U) \leq \gamma_R(T(v_0, R) - v_0, v_1 = 2)$. Similarly, $\gamma_R(U) \leq \gamma_R(T(v_0, R) - v_1, v_0 = 2)$. So, $\gamma_R(U) \leq \gamma$.

Let f be a $\gamma_R(U)$-function. We have $(f(v_0), f(v_1)) \in \{0,1,2\} \times \{0,1,2\} \setminus \{(0,2),(2,0)\}$ or $(f(v_0), f(v_1)) \in \{(0,2),(2,0)\}$. In the following we consider these cases.

- Let $(f(v_0), f(v_1)) = (a,b)$, where $(a,b) \in \{0,1,2\} \times \{0,1,2\} \setminus \{(0,2),(2,0)\}$. Function f is a RDF on $T(v_0,R)$ with $f(v_0) = a$ and $f(v_1) = b$ and so $\gamma_R(T(v_0,R), v_0 = a, v_1 = b) \leq \gamma_R(U)$.
- Let $(f(v_0), f(v_1)) = (2,0)$. The restriction of f to $V(U) \setminus \{v_1\}$ is a RDF on $U - v_1 = T(v_0,R) - v_1$ with $f(v_0) = 2$ and so $\gamma_R(T(v_0,R) - v_1, v_0 = 2) \leq \gamma_R(U)$.
- Similar to the previous case, if $(f(v_0), f(v_1)) = (0,2)$, then $\gamma_R(T(v_0,R) - v_0, v_1 = 2) \leq \gamma_R(U)$. So, $\gamma \leq \gamma_R(U)$.

This completes the proof. □

By Lemma 4.3 for computing the Roman domination number of a given unicyclic graph we need to compute the value $\gamma_R(T, u = a, v = b)$, where T is a tree with $u, v \in V(T)$ and $(a,b) \in \{0,1,2\} \times \{0,1,2\} \setminus \{(0,2),(2,0)\}$. We claim that Algorithms 4.2, 4.3 and 4.4 compute these values.

Lemma 4.4. *Let T be a rooted tree with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$ and let $(\gamma_{00}, \gamma'_{00}, \gamma_{01}, \gamma_{02})$ be the output of Algorithm* **RD0**(T,u,v). *Then,*

- $\gamma_{00} = \gamma_R(T, u = 0, v = 0)$,
- $\gamma'_{00} = \gamma_R(T, u, w, v = 0)$,
- $\gamma_{01} = \gamma_R(T, u = 1, v = 0)$,
- $\gamma_{02} = \gamma_R(T, u = 2, v = 0)$.

Proof. Let $P(T,v,u) = w_0(= v), \ldots, w_k(= u)$ $(k > 0)$ be the shortest path between v and u in T. The proof is by induction on $k = |P(T,v,u)|$. Let $k = 1$. So, u is the parent of v. Let $T' = T_u - T_v$. So,

- $\gamma_R(T, u = 0, v = 0) = \gamma_R(T_v, v = 0) + \gamma_R(T', u = 0)$,
- $\gamma_R(T, u, w, v = 0) = \gamma_R(T_v, v = 0) + \gamma_R(T' - u) + 2$,
- $\gamma_R(T, u = 1, v = 0) = \gamma_R(T_v, v = 0) + \gamma_R(T', u = 1)$,
- $\gamma_R(T, u = 2, v = 0) = \gamma_R(T_v - v) + \gamma_R(T', u = 2)$.

Since $k = 1$, the **for** loop of Algorithm **RD0**(T,u,v) does not execute. This proves the base case of the induction. Assume that the result is true for any rooted tree T' with the root u, $v \in V(T')$, a vertex $w \notin V(T)$ and $|P(T',v,u)| \leq m$, where $m \geq 1$. Let T be a rooted tree with the root u, $v \in V(T)$, a vertex $w \notin V(T)$ and $P(T,v,u) = w_0(= v), \ldots, w_m, w_{m+1}(= u)$. Let $(\gamma_0^i, \gamma_1^i, \gamma_2^i, \gamma_3^i)$ be values of variables $(\gamma_{00}, \gamma'_{00}, \gamma_{01}, \gamma_{02})$ of Algorithm **RD0**(T,u,v), respectively, after the iteration of the **for** loop for each $2 \leq i \leq m+1$. Let T_{w_m}

Algorithm 4.2: RD0(T,u,v)

Input: A connected rooted tree T with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$.

Output: $(\gamma_R(T,u=0,v=0), \gamma_R(T,u,w,v=0), \gamma_R(T,u=1,v=0), \gamma_R(T,u=2,v=0))$.

Let $P(T,v,u) = w_0(=v),\dots,w_k(=u)$ $(k>0)$ be the shortest path between u and v in T.
$T' = T_{w_1} - T_{w_0}$;
$\gamma_{00} = \gamma_R(T_{w_0}, w_0 = 0) + \gamma_R(T', w_1 = 0)$;
$\gamma'_{00} = \gamma_R(T_{w_0}, w_0 = 0) + \gamma_R(T' - w_1) + 2$;
$\gamma_{01} = \gamma_R(T_{w_0}, w_0 = 0) + \gamma_R(T', w_1 = 1)$;
$\gamma_{02} = \gamma_R(T_{w_0} - w_0) + \gamma_R(T', w_1 = 2)$;
for $i = 2$ **to** k **do**
  $T' = T_{w_i} - T_{w_{i-1}}$;
  $\alpha_0 = \min\{\gamma_R(T', w_i = 0) + \gamma_{00}, \gamma_R(T', w_i = 0) + \gamma_{01}, \gamma_R(T' - w_i) + \gamma_{02}\}$;
  $\alpha_1 = \gamma_R(T' - w_i) + \min\{\gamma_{00}, \gamma_{01}, \gamma_{02}\} + 2$;
  $\alpha_2 = \gamma_R(T', w_i = 1) + \min\{\gamma_{00}, \gamma_{01}, \gamma_{02}\}$;
  $\gamma_{02} = \gamma_R(T', w_i = 2) + \min\{\gamma'_{00} - 2, \gamma_{01}, \gamma_{02}\}$;
  $\gamma_{00} = \alpha_0$;
  $\gamma'_{00} = \alpha_1$;
  $\gamma_{01} = \alpha_2$;
return $(\gamma_{00}, \gamma'_{00}, \gamma_{01}, \gamma_{02})$;

be the rooted subtree of T with the root w_m. Let $(\alpha_{00}, \alpha'_{00}, \alpha_{01}, \alpha_{02})$ and $(\beta_{00}, \beta'_{00}, \beta_{01}, \beta_{02})$ be outputs of Algorithms **RD0**(T,u,v) and **RD0**(T_{w_m}, w_m, v), respectively. Clearly, $(\alpha_{00}, \alpha''_{00}, \alpha_{01}, \alpha_{02}) = (\gamma_0^{m+1}, \gamma_1^{m+1}, \gamma_2^{m+1}, \gamma_3^{m+1})$ and $(\beta_{00}, \beta'_{00}, \beta_{01}, \beta_{02}) = (\gamma_0^m, \gamma_1^m, \gamma_2^m, \gamma_3^m)$. By the induction hypothesis, we have $(\beta_{00}, \beta'_{00}, \beta''_{00}, \beta_{02}, \beta_{03}) = (\gamma_R(T_{w_m}, w_m = 0, v = 0), \gamma_R(T_{w_m}, w_m, w, v = 0), \gamma_R(T_{w_m}, w_m = 1, v = 0), \gamma_R(T_{w_m}, w_m = 2, v = 0))$.

Let $T' = T - T_{w_m}$. Since u is the parent of $w_m (\neq v)$ (i.e., u is adjacent to w_m) in T, we have

- $\gamma_R(T, u = 0, v = 0) = \min\{\gamma_R(T', u = 0) + \beta_{00}, \gamma_R(T', u = 0) + \beta_{01}, \gamma_R(T' - u) + \beta_{02}\}$
- $\gamma_R(T, u, w, v = 0) = \min\{\gamma_R(T' - u) + \beta_{00} + 2, \gamma_R(T' - u) + \beta_{01} + 2, \gamma_R(T' - u) + \beta_{02} + 2\}$
- $\gamma_R(T, u = 1, v = 0) = \min\{\gamma_R(T', u = 1) + \beta_{00}, \gamma_R(T', u = 1) + \beta_{01}, \gamma_R(T', u = 1) + \beta_{02}\}$

Algorithm 4.3: RD1(T, u, v)

Input: A connected rooted tree T with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$.

Output: $(\gamma_R(T, u = 0, v = 1), \gamma_R(T, u, w, v = 1), \gamma_R(T, u = 1, v = 1), \gamma_R(T, u = 2, v = 1))$.

Let $P(T, v, u) = w_0(= v), \ldots, w_k(= u)$ $(k > 0)$ be the shortest path between u and v in T.
$T' = T_{w_1} - T_{w_0}$;
$\gamma_{10} = \gamma_R(T_{w_0}, w_0 = 1) + \gamma_R(T', w_1 = 0)$;
$\gamma'_{10} = \gamma_R(T_{w_0}, w_0 = 1) + \gamma_R(T' - w_1) + 2$;
$\gamma_{11} = \gamma_R(T_{w_0}, w_0 = 1) + \gamma_R(T', w_1 = 1)$;
$\gamma_{12} = \gamma_R(T_{w_0}, w_0 = 1) + \gamma_R(T', w_1 = 2)$;
for $i = 2$ **to** k **do**
    $T' = T_{w_i} - T_{w_{i-1}}$;
    $\alpha_0 = \min\{\gamma_R(T', w_i = 0) + \gamma_{10}, \gamma_R(T', w_i = 0) + \gamma_{11}, \gamma_R(T' - w_i) + \gamma_{12}\}$;
    $\alpha_1 = \gamma_R(T' - w_i) + \min\{\gamma_{10}, \gamma_{11}, \gamma_{12}\} + 2$;
    $\alpha_2 = \gamma_R(T', w_i = 1) + \min\{\gamma_{10}, \gamma_{11}, \gamma_{12}\}$;
    $\gamma_{12} = \gamma_R(T', w_i = 2) + \min\{\gamma'_{10} - 2, \gamma_{11}, \gamma_{12}\}$;
    $\gamma_{10} = \alpha_0$;
    $\gamma'_{10} = \alpha_1$;
    $\gamma_{11} = \alpha_2$;
return $(\gamma_{10}, \gamma'_{10}, \gamma_{11}, \gamma_{12})$;

- $\gamma_R(T, u = 2, v = 0) = \min\{\gamma_R(T', u = 2) + \beta'_{00} - 2, \gamma_R(T', u = 2) + \beta_{01}, \gamma_R(T', u = 2) + \beta_{02}\}$.

This completes the proof. □

Similar to Lemma 4.4 we have the following results.

Lemma 4.5. *Let T be a rooted tree with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$ and let $(\gamma_{10}, \gamma'_{10}, \gamma_{11}, \gamma_{12})$ be the output of Algorithm* **RD1**(T, u, v). *Then,*

- $\gamma_{10} = \gamma_R(T, u = 0, v = 1)$,
- $\gamma'_{10} = \gamma_R(T, u, w, v = 1)$,
- $\gamma_{11} = \gamma_R(T, u = 1, v = 1)$,
- $\gamma_{12} = \gamma_R(T, u = 2, v = 1)$.

Lemma 4.6. *Let T be a rooted tree with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$ and let $(\gamma_{20}, \gamma'_{20}, \gamma_{21}, \gamma_{22})$ be the output of Algorithm* **RD2**(T, u, v). *Then,*

Algorithm 4.4: RD2(T, u, v)

Input: A connected rooted tree T with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$.

Output: $(\gamma_R(T, u = 0, v = 2), \gamma_R(T, u, w, v = 2), \gamma_R(T, u = 1, v = 2), \gamma_R(T, u = 2, v = 2))$.

Let $P(T, v, u) = w_0(= v), \ldots, w_k(= u)$ $(k > 0)$ be the shortest path between u and v in T.

$T' = T_{w_1} - T_{w_0}$;

$\gamma_{20} = \gamma_R(T_{w_0}, w_0 = 2) + \gamma_R(T' - w_1)$;

$\gamma'_{20} = \gamma_R(T_{w_0}, w_0 = 2) + \gamma_R(T' - w_1) + 2$;

$\gamma_{21} = \gamma_R(T_{w_0}, w_0 = 2) + \gamma_R(T', w_1 = 1)$;

$\gamma_{22} = \gamma_R(T_{w_0}, w_0 = 2) + \gamma_R(T', w_1 = 2)$;

for $i = 2$ **to** k **do**

> $T' = T_{w_i} - T_{w_{i-1}}$;
>
> $\alpha_0 = \min\{\gamma_R(T', w_i = 0) + \gamma_{20}, \gamma_R(T', w_i = 0) + \gamma_{21}, \gamma_R(T' - w_i) + \gamma_{22}\}$;
>
> $\alpha_1 = \gamma_R(T' - w_i) + \min\{\gamma_{20}, \gamma_{21}, \gamma_{22}\} + 2$;
>
> $\alpha_2 = \gamma_R(T', w_i = 1) + \min\{\gamma_{20}, \gamma_{21}, \gamma_{22}\}$;
>
> $\gamma_{22} = \gamma_R(T', w_i = 2) + \min\{\gamma'_{20} - 2, \gamma_{21}, \gamma_{22}\}$;
>
> $\gamma_{20} = \alpha_0$;
>
> $\gamma'_{20} = \alpha_1$;
>
> $\gamma_{21} = \alpha_2$;

return $(\gamma_{20}, \gamma'_{20}, \gamma_{21}, \gamma_{22})$;

- $\gamma_{20} = \gamma_R(T, u = 0, v = 2)$,
- $\gamma'_{20} = \gamma_R(T, u, w, v = 2)$,
- $\gamma_{21} = \gamma_R(T, u = 1, v = 2)$,
- $\gamma_{22} = \gamma_R(T, u = 2, v = 2)$.

Theorem 4.7. *There is a linear algorithm that computes the Roman domination number of a given unicyclic graph.*

Proof. Let U be a connected unicyclic graph with the unique cycle $v_0, \ldots, v_{k-1}, v_0$. By Lemma 4.3, $\gamma_R(U) = \min\{\gamma_R(T(v_0, R), v_0 = a, v_1 = b), \gamma_R(T(v_0, R) - v_0, v_1 = 2), \gamma_R(T(v_0, R) - v_1, v_0 = 2)\}$, where $(a, b) \in \{0, 1, 2\} \times \{0, 1, 2\} \setminus \{(0, 2), (2, 0)\}$. It follows from Lemmas 4.2, 4.4, 4.5 and 4.6 that we can compute $\gamma_R(U)$ using the outputs of Algorithms 4.1, 4.2, 4.3 and 4.4.

By Lemma 4.2 the running of Algorithm 4.1 is linear. It remains to compute running times of Algorithms 4.2, 4.3 and 4.4. Let T be a tree with $u, v \in V(T)$ and let $P(T, v, u) = w_0(= v), \ldots, w_k(= u)$ $(k > 0)$ be

Algorithm 5.1: D(T)

Input: A connected rooted tree $T = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ and Property 1.
Output: $(\gamma(T, v_1 = 0), \gamma(T, v_1 = 1), \gamma(T - v_1))$.

for $i = 1$ **to** n **do**
  $\gamma(v_i = 0) = \infty$;
  $\gamma(v_i = 1) = 1$;
  $\gamma(v_i) = 0$;
for $i = n$ **to** 2 **do**
  Let v_j be the parent of v_i;
  $\gamma(v_j = 0) = \min\{\gamma(v_j = 0) + \gamma(v_i = 0), \gamma(v_j) + \gamma(v_i = 1)\}$;
  $\gamma(v_j = 1) = \gamma(v_j = 1) + \min\{\gamma(v_i), \gamma(v_i = 1)\}$;
  $\gamma(v_j) = \gamma(v_j) + \min\{\gamma(v_i = 0) + \gamma(v_i = 1)\}$;
return $(\gamma(v_1 = 0), \gamma(v_1 = 1), \gamma(v_1))$;

the shortest path between u and v in T. Clearly, we can compute the rooted tree T_u with the root u for T and $P(T, v, u)$ in linear time. Let T_m be the value of the variable T' of Algorithm **RD0**(T, u, v) after the iteration of the **for** loop for each $2 \leq m \leq k$. Since the running time of Algorithm 4.1 is linear, the running time of lines 2-6 of Algorithm **RD0**(T, u, v) is $\mathcal{O}(V(T_1))$ and the running time of the iteration of the **for** loop of Algorithm **RD0**(T, u, v) for $2 \leq m \leq k$ is $\mathcal{O}(V(T_m))$. Clearly, $V(T_i) \cap V(T_j) = \emptyset$ for each $2 \leq i < j \leq k$. So, the running time of Algorithm **RD0**(T, u, v) is equal to $\sum_{i=2}^{k} \mathcal{O}(V(T_m)) = \mathcal{O}(V(T))$. Similarly, running times of Algorithms **RD1**(T, u, v) and **RD2**(T, u, v) are linear. This completes the proof. □

5. Computing domination number of unicyclic graphs

In this section, we give a linear algorithm that computes the domination number of unicyclic graphs. Let $G = (V, E)$ be a graph such that $u \in V$ and let $a \in \{0, 1\}$. We define the following.

- $\gamma(G, u = 0) = \min\{|S| : S \text{ is a DS on } G \text{ such that } u \notin S\}$,
- $\gamma(G, u = 1) = \min\{|S| : S \text{ is a DS on } G \text{ such that } u \in S\}$.

Similar to Lemma 4.1 we have the following.

Lemma 5.1. *Let $H_1 = (V_1, E_1)$ and $H_2 = (V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \emptyset$ such that $u \in V_1$ and $v \in V_2$. Let $G = (V_1 \cup V_2, E_1 \cup E_2 \cup \{uv\})$. Then, we have the following.*

Algorithm 5.2: D0(T, u, v)

Input: A connected rooted tree T with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$.

Output: $(\gamma(T, u = 0, v = 0), \gamma'(T, u, v = 0, w), \gamma(T, u = 1, v = 0))$.

Let $P(T, v, u) = w_0(= v), \ldots, w_k(= u)$ $(k > 0)$ be the shortest path between u and v in T.
$T' = T_{w_1} - T_{w_0}$;
$\gamma_{00} = \gamma(T_{w_0}, w_0 = 0) + \gamma(T', w_1 = 0)$;
$\gamma'_{00} = \gamma(T_{w_0}, w_0 = 0) + \gamma(T' - w_1) + 1$;
$\gamma_{01} = \gamma(T_{w_0} - w_0) + \gamma(T', w_1 = 1)$;
for $i = 2$ **to** k **do**
  $T' = T_{w_i} - T_{w_{i-1}}$;
  $\alpha_0 = \min\{\gamma(T', w_i = 0) + \gamma_{00}, \gamma(T' - w_i) + \gamma_{01}\}$;
  $\alpha_1 = \gamma(T' - w_i) + \min\{\gamma_{00}, \gamma_{01}\} + 1$;
  $\gamma_{01} = \gamma(T', w_i = 1) + \min\{\gamma'_{00} - 1, \gamma_{01}\}$;
  $\gamma_{00} = \alpha_0$;
  $\gamma'_{00} = \alpha_1$;
return $(\gamma_{00}, \gamma'_{00}, \gamma_{01})$;

(*i*) $\gamma(G, u = 0) = \min\{\gamma(H_1, u = 0) + \gamma(H_2, v = 0), \gamma(H_1 - u) + \gamma(H_2, v = 1)\}$,

(*ii*) $\gamma(G, u = 1) = \min\{\gamma(H_1, u = 1) + \gamma(H_2 - v), \gamma(H_1, u = 1) + \gamma(H_2, v = 1)\}$,

(*iii*) $\gamma(G - u) = \gamma(H_1 - u) + \min\{\gamma(H_2, v = 0), \gamma(H_2, v = 1)\}$.

Similar to Lemma 4.2, we have the following.

Lemma 5.2. *Let T be a tree with $u \in V$. Algorithm 5.1 computes values $\gamma(T, u = 0)$, $\gamma(T, u = 1)$ and $\gamma(T - u)$ in linear time.*

Let $G = (V, E)$ be a graph with $u, v \in V$ and a vertex $w \notin V$. We define the following.

- $\gamma(G, u = 0, v = 0) = \min\{|S| : S$ is a DS on G such that $u \notin S$ and $v \notin S\}$,
- $\gamma(G, u = 0, v = 1) = \min\{|S| : S$ is a DS on G such that $u \notin S$ and $v \in S\}$,
- $\gamma(G, u = 1, v = 0) = \min\{|S| : S$ is a DS on G such that $u \in S$ and $v \notin S\}$,

Algorithm 5.3: D1(T, u, v)

Input: A connected rooted tree T with the root u, $v \in V(T)$ and a vertex $w \notin V(T)$.

Output: $(\gamma(T, u = 0, v = 1), \gamma'(T, u, v = 1, w), \gamma(T, u = 1, v = 1))$.

Let $P(T, v, u) = w_0(= v), \ldots, w_k(= u)$ $(k > 0)$ be the shortest path between u and v in T.
$T' = T_{w_1} - T_{w_0}$;
$\gamma_{10} = \gamma(T_{w_0}, w_0 = 1) + \gamma(T' - w_1)$;
$\gamma'_{10} = \gamma(T_{w_0}, w_0 = 1) + \gamma(T' - w_1) + 1$;
$\gamma_{11} = \gamma(T_{w_0}, w_0 = 1) + \gamma(T', w_1 = 1)$;
for $i = 2$ **to** k **do**
    $T' = T_{w_i} - T_{w_{i-1}}$;
    $\alpha_0 = \min\{\gamma(T', w_i = 0) + \gamma_{10}, \gamma(T' - w_i) + \gamma_{11}\}$;
    $\alpha_1 = \gamma(T' - w_i) + \min\{\gamma_{10}, \gamma_{11}\} + 1$;
    $\gamma_{11} = \gamma(T', w_i = 1) + \min\{\gamma'_{10} - 1, \gamma_{11}\}$;
    $\gamma_{10} = \alpha_0$;
    $\gamma'_{10} = \alpha_1$;
return $(\gamma_{10}, \gamma'_{10}, \gamma_{11})$;

- $\gamma(G, u = 1, v = 1) = \min\{|S| : S$ is a DS on G such that $u \in S$ and $v \in S\}$,
- $\gamma(G, u, w, v = 0) = \min\{|S| : S$ is a DS on $G + uw$ such that $w \in S$ and $u, v \notin S\}$,
- $\gamma(G, u, w, v = 1) = \min\{|S| : S$ is a DS on $G + uw$ such that $v, w \in S$ and $u \notin S\}$.

Let U be a connected unicyclic graph with the unique cycle $C = v_0, \ldots, v_{k-1}, v_0$, where $k \geq 3$. Recall that $T(v_0, R) = U - v_0v_1$. Similar to Lemma 4.3 we have the following.

Lemma 5.3. *Let U be a connected unicyclic graph with the unique cycle $v_0, \ldots, v_{k-1}, v_0$ $(k > 2)$. Then,* $\gamma(U) = \min\{\gamma(T(v_0, R), v_0 = 0, v_1 = 0), \gamma(T(v_0, R), v_0 = 1, v_1 = 1), \gamma(T(v_0, R) - v_1, v_0 = 1), \gamma(T(v_0, R) - v_0, v_1 = 1)\}$.

By Lemma 5.3 for computing the domination number of a given unicyclic graph we need to compute values $\gamma(T, u = 0, v = 0)$ and $\gamma(T, u = 1, v = 1)$, where T is a tree with $u, v \in V(T)$. We claim that Algorithms 5.2 and 5.3 compute these values. Similar to Lemma 4.4 we have the following results.

Lemma 5.4. *Let T be a rooted tree with the root u, $v \in V(T)$ and $w \notin V(T)$ and let $(\gamma_{00}, \gamma'_{00}, \gamma_{01})$ be the output of Algorithm* $\mathbf{D0}(T, u, v)$. *Then,*

- $\gamma_{00} = \gamma(T, u = 0, v = 0)$,
- $\gamma'_{00} = \gamma(T, u, w, v = 0)$,
- $\gamma_{01} = \gamma(T, u = 1, v = 0)$.

Lemma 5.5. *Let T be a rooted tree with the root u, let $v \in V(T)$ and $w \notin V(T)$ and let $(\gamma_{10}, \gamma'_{10}, \gamma_{11})$ be the output of Algorithm* $\mathbf{D1}(T, u, v)$. *Then,*

- $\gamma_{10} = \gamma(T, u = 0, v = 1)$,
- $\gamma'_{10} = \gamma(T, u, w, v = 1)$,
- $\gamma_{11} = \gamma(T, u = 1, v = 1)$.

Similar to Theorem 4.7 we have the following.

Theorem 5.6. *There is a linear algorithm that computes the domination number of a given unicyclic graph.*

By Theorems 4.7 and 5.6 we obtain the following.

Theorem 5.7. *There is a linear algorithm that decides whether a given unicyclic graph is a Roman graph.*

References

1. H. Abdollahzadeh Ahangar, M. Chellali, S. M. Sheikholeslami, On the Roman domination in graphs, *Discrete Appl. Math.*, **232** (2017), 1–7.
2. R. A. Beeler, T. W. Haynesa and S. T. Hedetniemi, Double Roman domination, *Discrete Appl. Math.*, **211** (2016), 23–29.
3. M. Chellali, T. W. Haynes, S. T. Hedetniemi, A. MacRae, Roman {2}-domination, *Discrete Appl. Math.*, **204** (2016), 22–28.
4. E. J. Cockayane, P. M. Dreyer Jr., S. M. Hedetniemi and S. T. Hedetniemi, On Roman domination in graphs, *Discrete Math.*, **278** (2004), 11–22.
5. N. Jafari Rad and H. Rahbani, Some progress on the Roman domination in graphs, *Discuss. Math. Graph Theory*, **39** (2019), 41–53.
6. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
7. T. W. Haynes, S. T. Hedetniemi and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, In c., New York, 1998.
8. M. A. Henning, A characterization of Roman trees, *Discuss. Math. Graph Theory*, **22** (2002), 325–334.
9. M. Liedloff, T. Kloks, J. Liu and S.-L. Pen, Efficient algorithms for Roman domination on some classes of graphs, *Discrete Appl. Math.*, **156** (2008), 3400–3415.
10. C. S. Revelle and K. E. Rosing, Defendens imperium romanum: a classical problem in military strategy, *Amer. Math. Monthly* **107** (2000), 585–594.

11. I. Stewart, Defend the roman empire!, *Sci. Amer.*, **281** (1999), 136–139.
12. J. Yue, M. Wei, M. Li and G. Liu, On the Roman domination of graphs, *Appl. Math. Comput.*, **338** (2018), 669–675.
13. X. Zhang, Z. Li, H. Jiang and Z. Shao, Double Roman domination in trees, *Inform. Process. Lett.*, **134** (2018), 31–34.

Abolfazl Poureidi
Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran.
Email: a.poureidi@shahroodut.ac.ir

Journal of Algebraic Systems

ALGORITHMIC ASPECTS OF ROMAN GRAPHS

A. POUREIDI

صورت‌های الگوریتمی گراف‌های رومن

ابوالفضل پورعیدی

دانشکده علوم ریاضی، دانشگاه صنعتی شاهرود، شاهرود، ایران

فرض کنید $G=(V,E)$ یک گراف است. مجموعه‌ی $S\subseteq V$ را یک مجموعه احاطه‌گر G می‌نامیم اگر هر راس در $V\setminus S$ مجاور به حداقل یک راس در S است. عدد احاطه‌گر G برابر با کمترین اندازه یک مجموعه احاطه‌گر G است که آن‌را با $\gamma(G)$ نمایش می‌دهیم. یک تابع احاطه‌گر رومن (RDF) برای G تابع $f:V\rightarrow\{۰,۱,۲\}$ است به طوری که هر راس $v\in V$ با $f(v)=۰$ مجاور به یک راس u با $f(u)=۲$ است. وزن f برابر با $f(V)=\sum_{v\in V}f(v)$ است. کمترین وزن یک RDF برای G را عدد احاطه‌گری رومن G می‌نامیم و آن‌را با $\gamma_R(G)$ نمایش می‌دهیم. گراف G را یک گراف رومن می‌نامیم اگر $\gamma_R(G)=۲\gamma(G)$.

در این مقاله ابتدا نشان می‌دهیم که مسئله تصمیم‌گیری در مورد اینکه یک گراف رومن است یک مسئله NP-hard است. سپس یک الگوریتم زمان خطی ارائه می‌کنیم که تصمیم می‌گیرد یک گراف تک‌دور یک گراف رومن است.

کلمات کلیدی: مجموعه احاطه‌گر، تابع احاطه‌گر رومن، الگوریتم.