THE SIGNED ROMAN DOMINATION NUMBER OF LADDER GRAPHS AND THEIR COMPLEMENTS

DILBAK HAJE, DELBRIN AHMED, HASSAN IZANLOO, AND MANJIL SAIKIA

ABSTRACT. A *signed Roman dominating function* (SRDF) on $G = (V, E)$, a finite, connected, simple graph is a mapping

$$f : V \to \{-1, 1, 2\}$$

such that

(1) For every vertex $x \in V$,

$$\sum_{y \in N[x]} f(y) \geq 1,$$

where $N[x]$ denotes the closed neighborhood of x, consisting of x together with all vertices adjacent to x.

(2) Every vertex $x \in V$ with $f(x) = -1$ is adjacent to at least one vertex $y \in V$ such that $f(y) = 2$.

The *weight* of an SRDF is defined as $\sum_{v \in V(G)} f(v)$. The *signed Roman domination number* (SRDN) of G, denoted by $\gamma_{SR}(G)$, is the minimum possible weight among all signed Roman dominating functions on G.

In this work, we determine the signed Roman domination number of the ladder graph LG_n and its complement LG_n^c.

1. Introduction & Motivation

Let G be a simple, connected graph with vertex set $V(G)$ and edge set $E(G)$, where $|V(G)| = n$ and $|E(G)| = m$. We restrict our investigation to non-trivial graphs. The set of neighbors of a vertex u, denoted by $N_G(u) = \{v : uv \in E(G)\}$, is referred to as the open neighborhood, whereas $N_G[u] = N_G(u) \cup \{u\}$ constitutes the closed neighborhood. We denote the degree of u by $deg_G(u)$, and let $\delta(G)$ and $\Delta(G)$ represent the minimum and maximum degrees among all vertices. A graph is termed k-regular when every vertex has degree k (i.e., $\delta(G) = \Delta(G) = k$). Regarding specific graph classes, we let K_n, P_n, and C_n represent the complete graph, path, and cycle on n vertices, respectively. Finally, the complement of G, denoted G^c, consists of the vertex set $V(G)$ and an edge set containing exactly those pairs of distinct vertices not present in $E(G)$.

A subset $D \subseteq V$ is defined as a *dominating set* if every vertex in $V \setminus D$ is adjacent to at least one vertex in D. The *domination number*, denoted by $\gamma(G)$, represents the minimum cardinality among all such sets. The study of domination theory has its roots in the seminal works of Ore [Ore62] and Berge [Ber73]. Furthermore, it is well-established that computing the domination number for general graphs is an NP-complete problem, a result attributed to Garey and Johnson [GJ79].

A function $f : V(G) \to \{0, 1\}$ is called a *dominating function* on G if for each vertex $u \in V(G)$, $\sum_{v \in N_G[u]} f(v) \geq 1$. The value $w(f) = \sum_{u \in V(G)} f(u)$ is called the *weight* of f. Thus, $\gamma(G)$

2020 *Mathematics Subject Classification.* 05C69, 05C78, 90C27.

Key words and phrases. Signed Roman Domination number, dominating function.

Corresponding Author: Manjil Saikia (manjil@saikia.in).

is the minimum value of $w(f)$. In a similar vein, we define a *signed dominating function* on a graph G as an assignment of $+1$ and -1 to the vertices, such that the sum of labels within every closed neighborhood is positive. The signed domination number is defined as the minimum total weight taken over all such valid functions. The primary focus of this work, however, is a variation known as the signed Roman domination function, which is defined below.

Definition 1. Let $G = (V, E)$ be a graph. A *signed Roman domination function* (SRDF) on the graph G is a function $f : V(G) \to \{-1, 1, 2\}$ such that:

(i) For each $u \in V(G)$, $\sum_{v \in N_G[u]} f(v) \geq 1$, and

(ii) Each vertex u for which $f(u) = -1$ is adjacent to at least one vertex v for which $f(v) = 2$.

The *weight* of f, denoted by $w(f)$ is the value $f(V) = \sum_{u \in V(G)} f(u)$. The *signed Roman domination number* (SRDN) of G, $\gamma_{SR}(G)$, is the minimum weight of an SRDF on G.

Ahangar et al. [AAHL+14] introduced the concept of an SRDF. Clearly, $\gamma_{SR}(G) \leq n$ as we can just assign $+1$ to all vertices yielding a valid SRDF. In [AAHL+14] Ahangar et al. presented various lower and upper bounds on the SRDN of a graph in terms of its order, size and vertex degrees. For instance, they showed that for a graph G with n vertices, we have

$$\gamma_{SR}(G) \geq \left(\frac{-2\Delta^2 + 2\Delta\delta + \Delta + 2\delta + 3}{(\Delta + 1)(2\Delta + \delta + 3)} \right) n. \tag{1.1}$$

Moreover, they also showed that if G is a graph with n vertices and m edges without isolated vertex, then we have

$$\gamma_{SR}(G) \geq \frac{3n - 4m}{2}. \tag{1.2}$$

They also investigated the SRDN of some special bipartite graphs.

Observe that every SRDF f on a graph G is characterized by an ordered partition (V_{-1}, V_1, V_2) of the vertex set $V(G)$, where $V_i = \{u \in V(G) \mid f(u) = i\}$ for each $i \in \{-1, 1, 2\}$. Consequently, the weight of the function is given by $w(f) = 2|V_2| + |V_1| - |V_{-1}|$. For the sake of brevity, we identify f with the triplet (V_{-1}, V_1, V_2) and denote the cumulative weight of a subset $U \subseteq V(G)$ as $f(U) = \sum_{u \in U} f(u)$. An SRDF that achieves the minimum weight, i.e., $w(f) = \gamma_{SR}(G)$, is referred to as a $\gamma_{SR}(G)$*-function* or an *optimal SRDF*.

Several follow-up work has been performed on SRDFs after the work of Ahangar et al. [AAHL+14]. For instance, Behtoei, Vatandoost and Azizi [BVA16] studied the SRDN of the join of graphs, and determined its value for the join of cycles, wheels, fans, and friendship graphs; while, Hong et. al. [HYZZ20] determined its value for spider graphs and double-star graphs.

In this paper, we investigate the SRDN of the Ladder graph LG_n and their complement graphs LG_n^C. The paper is arranged as follows: in Section 2 we evaluate the SRDN for LG_n and in Section 3 we evaluate the SRDN for the complement of LG_n.

2. SRDN for Ladder graphs LG_n

The *Cartesian product* of two graphs G_1 and G_2, denoted by $G_1 \times G_2$, is the graph defined on the vertex set $V(G_1) \times V(G_2)$. Two vertices (u_1, v_1) and (u_2, v_2) are adjacent in $G_1 \times G_2$ if and only if $u_1 = u_2$ and $v_1 v_2 \in E(G_2)$, or $v_1 = v_2$ and $u_1 u_2 \in E(G_1)$.

In this section, we are interested in the following cartesian product.

Definition 2. The *Ladder graph* of order $2n$, denoted by LG_n, is given by $LG_n := P_2 \times P_n$, where P_i is the path graph on i vertices.

Note that by the definition of LG_n, one can write

$V(LG_n) = \{(1,k),(2,k) | \, k \in [n]\}$, and

$E(LG_n) = \{\{(1,k),(2,k)\} | \, 1 \leq k \leq n\} \cup \{\{(1,k),(1,k+1)\},\{(2,k),(2,k+1)\} | \, 1 \leq k \leq n-1\}.$

For simplicity, we encode the vertices $(1,i)$ and $(2,i)$ with $1i$ and $2i$ for $i = 1,2,\dots,n$.

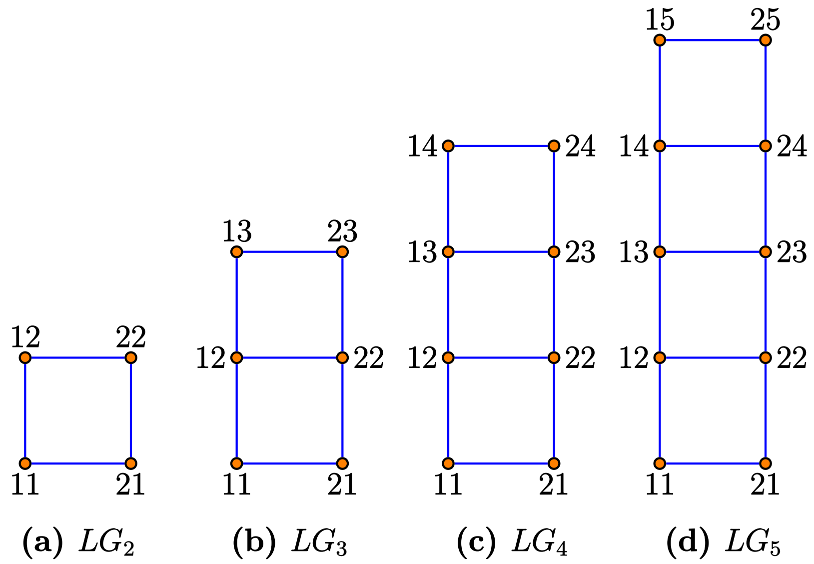

FIGURE 1. The ladder graphs LG_2, LG_3, LG_4, and LG_5.

From the definition, we have

(i) $|E(LG_n)| = 3n-2$, and

(ii) LG_n has four vertices of degree 2 and $2n-4$ vertices of degree 3.

We now proceed to find the SRDN for Ladder graphs.

From (1.1) we get a bound of

$$\gamma_{SR}(LG_n) \geq \frac{n}{2}.$$

We improve this in the next result.

Theorem 1. *Let LG_n be the Ladder graph of order $2n$ with $n \geq 2$. Then*

$$\gamma_{SR}(LG_n) = \left\lfloor \frac{n+2}{2} \right\rfloor + 1.$$

Proof. First we show that

$$\gamma_{SR}(LG_n) \leq \left\lfloor \frac{n+2}{2} \right\rfloor + 1,$$

by exhibiting a function which satisfies the bound.

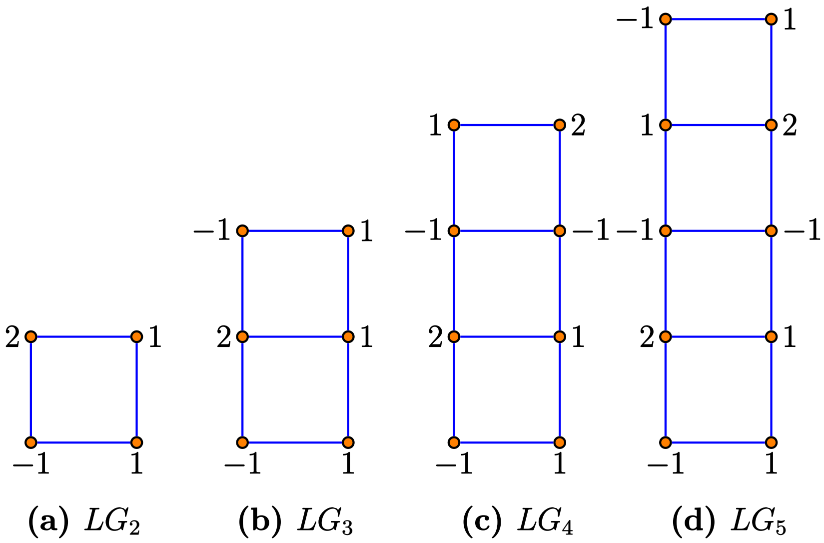

FIGURE 2. SRDF labelings of LG_2, LG_3, LG_4, and LG_5.

We define a function

$$f : V(LG_n) \longrightarrow \{-1, 1, 2\}$$

as follows. For the first column, set

$$f(1,i) = \begin{cases} -1, & i \equiv 1,3 \pmod 4, \text{ except that } f(1,n) = 1 \text{ if } n \equiv 1 \pmod 4, \\ 1, & i \equiv 0 \pmod 4, \\ 2, & i \equiv 2 \pmod 4. \end{cases}$$

For the second column, set

$$f(2,i) = \begin{cases} 1, & i = 1, \\ 1, & i \equiv 2 \pmod 4, \\ 1, & i = n \text{ and } n \equiv 3 \pmod 4, \\ 2, & i \equiv 0 \pmod 4, \\ -1, & i \equiv 1,3 \pmod 4,\ i \geq 3, \text{ except when } i = n \text{ and } n \equiv 3 \pmod 4. \end{cases}$$

We first verify that f is an SRDF.

Let $v = (1, i)$ with $f(v) = -1$. Then i is odd. If $1 < i < n$, then the two vertical neighbors of v have even indices $i - 1$ and $i + 1$. One of these indices is congruent to $2 \pmod 4$, and hence the corresponding first-row neighbor has label 2. Therefore v has a neighbor labeled 2.

At $v = (1, 1)$, it has neighbor $(1, 2)$, and

$$f(1, 2) = 2.$$

At the other endpoint, if $f(1, n) = -1$, then n is odd and $n \not\equiv 1 \pmod 4$, so $n \equiv 3 \pmod 4$. Thus $n - 1 \equiv 2 \pmod 4$, and

$$f(1, n - 1) = 2.$$

Hence every first-column vertex labeled -1 has a neighbor labeled 2.

Now let $v = (2, i)$ with $f(v) = -1$. Then i is odd, $i \geq 3$, and v is not the exceptional terminal vertex when $n \equiv 3 \pmod 4$. If $1 < i < n$, then the two vertical neighbors of v have indices $i-1$ and $i+1$, one of which is congruent to $0 \pmod 4$. The corresponding second-column neighbor therefore has label 2. If $i = n$, then necessarily $n \equiv 1 \pmod 4$, and so $n - 1 \equiv 0 \pmod 4$; hence

$$f(2, n-1) = 2.$$

Thus every vertex labeled -1 has a neighbor labeled 2. So, we have checked that f satisfies the second condition of being an SRDF.

We now check that f satisfies the first condition. Since every internal vertex of LG_n has degree 3 and the endpoints have degree 2, it suffices to check all the possible local configurations.

For first-column internal vertices with odd index i, we have

$$f(1, i) = -1, \qquad \{f(1, i-1), f(1, i+1)\} = \{1, 2\}, \qquad f(2, i) \geq -1.$$

Hence

$$\sum_{u \in N[(1,i)]} f(u) \geq -1 + 1 + 2 - 1 = 1.$$

Endpoint cases only remove one vertical neighbor or replace a -1 by 1, and the same inequality remains valid.

For second-column internal vertices with odd index i and $f(2, i) = -1$, similarly,

$$f(2, i) = -1, \qquad \{f(2, i-1), f(2, i+1)\} = \{1, 2\}, \qquad f(1, i) = -1,$$

so

$$\sum_{u \in N[(2,i)]} f(u) = -1 + 1 + 2 - 1 = 1.$$

If $i \equiv 0 \pmod 4$, then

$$f(1, i) = 1, \qquad f(2, i) = 2.$$

The two vertical neighbors in the same column have odd indices and therefore have labels at least -1. Hence

$$\sum_{u \in N[(1,i)]} f(u) \geq 1 + 2 - 1 - 1 = 1,$$

and

$$\sum_{u \in N[(2,i)]} f(u) \geq 2 + 1 - 1 - 1 = 1.$$

If $i \equiv 2 \pmod 4$, then

$$f(1, i) = 2, \qquad f(2, i) = 1.$$

Again the two vertical neighbors in the same column have labels at least -1, so

$$\sum_{u \in N[(1,i)]} f(u) \geq 2 + 1 - 1 - 1 = 1,$$

and

$$\sum_{u \in N[(2,i)]} f(u) \geq 1 + 2 - 1 - 1 = 1.$$

Therefore f is an SRDF on LG_n.

We now compute its weight. Let

$$V_j = \{v \in V(LG_n) : f(v) = j\}, \qquad j \in \{-1, 1, 2\}.$$

The vertices labeled 2 occur at

$$(1,i) \quad \text{with } i \equiv 2 \pmod 4,$$

and

$$(2,i) \quad \text{with } i \equiv 0 \pmod 4.$$

Hence

$$|V_2| = \left\lfloor \frac{n}{2} \right\rfloor.$$

The vertices labeled -1 occur at odd-indexed rows, with the following exceptions: $(2,1)$ is labeled 1 rather than -1, and if n is odd, exactly one terminal vertex is also changed from -1 to 1. Therefore

$$|V_{-1}| = n-1.$$

Since $|V(LG_n)| = 2n$, it follows that

$$|V_1| = 2n - |V_{-1}| - |V_2| = 2n - (n-1) - \left\lfloor \frac{n}{2} \right\rfloor = \left\lceil \frac{n}{2} \right\rceil + 1.$$

Thus

$$\begin{aligned} \omega(f) &= 2|V_2| + |V_1| - |V_{-1}| \\ &= 2\left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil + 1 - (n-1) \\ &= \left\lfloor \frac{n}{2} \right\rfloor + 2 \\ &= \left\lfloor \frac{n+2}{2} \right\rfloor + 1. \end{aligned}$$

Therefore

$$\gamma_{\mathrm{sR}}(LG_n) \le \left\lfloor \frac{n+2}{2} \right\rfloor + 1.$$

We now prove the reverse inequality

$$\gamma_{\mathrm{sR}}(LG_n) \ge \left\lfloor \frac{n}{2} \right\rfloor + 2.$$

Let

$$f : V(LG_n) \longrightarrow \{-1, 1, 2\}$$

be an arbitrary SRDF. For each row i, define the row labelling

$$L_i = (f(1,i), f(2,i)).$$

There are exactly nine possible row labellings:

labelling	A	B	C	D	E	F	G	H	I
$(f(1,i), f(2,i))$	$(-1,-1)$	$(-1,1)$	$(-1,2)$	$(1,-1)$	$(1,1)$	$(1,2)$	$(2,-1)$	$(2,1)$	$(2,2)$

and their weights are

labelling	A	B	C	D	E	F	G	H	I
w	-2	0	1	0	2	3	1	3	$4.$

A triple of row labellings (P, Q, R) is called *admissible* if the two vertices in the middle row Q satisfy both SRDF conditions using the three consecutive row labellings P, Q, R. More explicitly, if

$$P = (p_1, p_2), \qquad Q = (q_1, q_2), \qquad R = (r_1, r_2),$$

then (P, Q, R) is admissible if and only if

$$p_1 + q_1 + r_1 + q_2 \geq 1, \qquad p_2 + q_2 + r_2 + q_1 \geq 1,$$

and, whenever $q_j = -1$, at least one of the three neighbours of that vertex has label 2.

Similarly, a pair (Q, R) is called *bottom-admissible* if the first row labelling Q satisfies both SRDF conditions using only the row-neighbour and the right row labelling R. Thus, if $Q = (q_1, q_2)$ and $R = (r_1, r_2)$, then

$$q_1 + q_2 + r_1 \geq 1, \qquad q_2 + q_1 + r_2 \geq 1,$$

and, whenever $q_j = -1$, at least one of the three neighbours of that vertex has label 2. Top-admissibility is defined analogously.

Let M_n be the minimum possible weight of a sequence

$$L_1, L_2, \ldots, L_n$$

of n row labellings such that

(1) (L_1, L_2) is bottom-admissible;

(2) (L_{i-1}, L_i, L_{i+1}) is admissible for every $2 \leq i \leq n-1$;

(3) (L_{n-1}, L_n) is top-admissible.

Every SRDF on LG_n determines such an admissible sequence of row labellings. Therefore

$$\gamma_{\text{sR}}(LG_n) \geq M_n.$$

We now compute M_n. For $i \geq 2$ and row labellings P, Q, let $D_i(P, Q)$ be the minimum weight of an admissible partial sequence of length i ending in the two row labellings P, Q, where all rows up to row $i-1$ have already been checked. If no such partial sequence exists, set $D_i(P, Q) = +\infty$.

The initial condition is

$$D_2(P, Q) = \begin{cases} w(P) + w(Q), & (P, Q) \text{ is bottom-admissible,} \\ +\infty, & \text{otherwise.} \end{cases}$$

The recurrence is

$$D_{i+1}(Q, R) = w(R) + \min_{\substack{P \\ (P,Q,R) \text{ admissible}}} D_i(P, Q).$$

Finally,

$$M_n = \min\{D_n(P, Q) : (P, Q) \text{ is top-admissible}\}.$$

The finite verification in Appendix A[1] gives

$$M_2 = 3, \qquad M_3 = 3, \qquad M_4 = 4, \qquad M_5 = 4,$$

and, for every $n \geq 6$,

$$M_n = M_{n-2} + 1.$$

[1]We have used ChatGPT in this verification, and then human verified the output.

The verification consists of checking the 9^3 possible internal triples and the 9^2 possible endpoint pairs. Hence,

$$M_{2k} = k + 2, \qquad M_{2k+1} = k + 2.$$

Therefore, for every $n \geq 2$,

$$M_n = \left\lfloor \frac{n}{2} \right\rfloor + 2.$$

Since every SRDF on LG_n has weight at least M_n, we obtain

$$\gamma_{\mathrm{sR}}(LG_n) \geq \left\lfloor \frac{n}{2} \right\rfloor + 2.$$

Combining this lower bound with the construction above gives

$$\gamma_{\mathrm{sR}}(LG_n) = \left\lfloor \frac{n}{2} \right\rfloor + 2 = \left\lfloor \frac{n+2}{2} \right\rfloor + 1.$$

□

3. SRDN for the Complement of the Ladder Graphs

Now, we turn our attention to LG_n^C and determine their SRDN. As an illustration, the complement LG_n^C when $n = 2$ and 3 is shown in Figure 3 with $\gamma_{SR}(LG_2^C) = 2$ and $\gamma_{SR}(LG_3^C) = 3$.

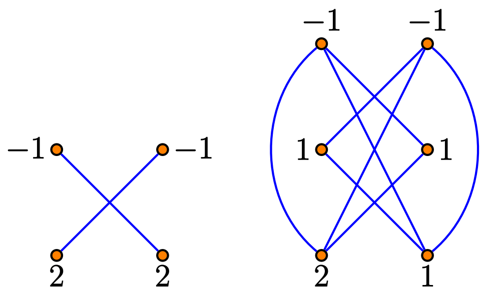

Figure 3. SRDF for LG_2^C and LG_3^C.

From (1.1), we get the bound of $\gamma_{SR}(LG_n^C) > 1$ for $n \geq 2$. We improve this bound below.

Theorem 2. *Let LG_n^C be complement of the Ladder graph. Then, for all $n \geq 4$, we have*

$$\gamma_{SR}(LG_n^C) = 2.$$

Proof. Let

$$V(LG_n^C) = \{(1,k),(2,k) : 1 \leq k \leq n\},$$

where the vertices $(1,k)$ and $(2,k)$ lie in the first and second rows[2], respectively. Define $f : V(LG_n^C) \to \{-1, 1, 2\}$ by

$$f(v) = \begin{cases} -1, & \text{if } v = (1,k),\ 2 \leq k \leq n-1, \text{ or } v = (2,2),(2,n-1), \\ 1, & \text{if } v = (2,k),\ k \neq 2, n-1, \\ 2, & \text{if } v = (1,1) \text{ or } v = (1,n). \end{cases}$$

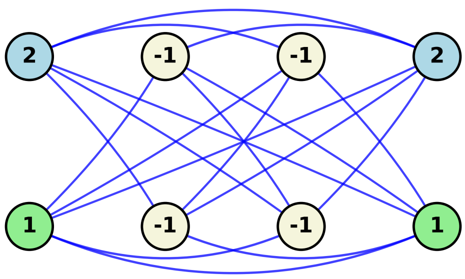

FIGURE 4. An SRDF labeling of LG_4^C, where blue edges indicate adjacency and vertex labels $2, 1$, and -1 represent the assigned signed Roman domination values.

As an illustration of how this labeling function works see Figure 4.

We first show that f is a signed Roman dominating function on LG_n^C. Let

$$\omega(f) = \sum_{v \in V(LG_n^C)} f(v)$$

denote the weight of f. There are two vertices labelled 2, namely $(1,1)$ and $(1,n)$. There are $n-2$ internal vertices in the first row labelled -1, together with the two vertices $(2,2)$ and $(2,n-1)$ in the second row. Thus n vertices are labelled -1. The remaining $n-2$ vertices in the second row are labelled 1. Hence

$$\omega(f) = 2 \cdot 2 + (n-2) \cdot 1 - n = 4 + n - 2 - n = 2.$$

It remains to verify the two defining conditions of a signed Roman dominating function. We use the relation between the neighbourhoods in LG_n and in its complement. For every vertex $x \in V(LG_n^C)$, we have

$$N_{LG_n^C}[x] = V(LG_n) \setminus N_{LG_n}(x),$$

where $N_{LG_n}(x)$ is the open neighbourhood of x in the original ladder graph. Therefore

$$\sum_{y \in N_{LG_n^C}[x]} f(y) = \omega(f) - \sum_{y \in N_{LG_n}(x)} f(y) = 2 - \sum_{y \in N_{LG_n}(x)} f(y).$$

Thus it is enough to show that

$$\sum_{y \in N_{LG_n}(x)} f(y) \leq 1$$

for every vertex x. We verify this according to the position of x. For the corner vertices in the first row,

$$\sum_{y \in N_{LG_n}((1,1))} f(y) = f((1,2)) + f((2,1)) = -1 + 1 = 0,$$

and similarly

$$\sum_{y \in N_{LG_n}((1,n))} f(y) = f((1,n-1)) + f((2,n)) = -1 + 1 = 0.$$

[2]From hereon we rotate the graph for notational ease, see Figure 4 for an example.

For the corner vertices in the second row,

$$\sum_{y\in N_{LG_n}((2,1))} f(y) = f((2,2)) + f((1,1)) = -1+2 = 1,$$

and

$$\sum_{y\in N_{LG_n}((2,n))} f(y) = f((2,n-1)) + f((1,n)) = -1+2 = 1.$$

Hence the required inequality holds for all four corner vertices. Now consider an internal vertex $(1,k)$ in the first row, where $2 \le k \le n-1$. Its neighbours in LG_n are

$$(1,k-1), \quad (1,k+1), \quad (2,k).$$

If $k=2$, then

$$f((1,1)) + f((1,3)) + f((2,2)) = 2-1-1 = 0.$$

Similarly, if $k=n-1$, then

$$f((1,n-2)) + f((1,n)) + f((2,n-1)) = -1+2-1 = 0.$$

For $3 \le k \le n-2$, we have

$$f((1,k-1)) + f((1,k+1)) + f((2,k)) \le -1-1+1 = -1.$$

Thus

$$\sum_{y\in N_{LG_n}((1,k))} f(y) \le 1$$

for every internal vertex in the first row. Next consider an internal vertex $(2,k)$ in the second row, where $2 \le k \le n-1$. Its neighbours in LG_n are

$$(2,k-1), \quad (2,k+1), \quad (1,k).$$

If $k=2$, then

$$f((2,1)) + f((2,3)) + f((1,2)) \le 1+1-1 = 1.$$

Similarly, if $k=n-1$, then

$$f((2,n-2)) + f((2,n)) + f((1,n-1)) \le 1+1-1 = 1.$$

For $3 \le k \le n-2$, we have

$$f((2,k-1)) + f((2,k+1)) + f((1,k)) \le 1+1-1 = 1.$$

Therefore, for every vertex $x \in V(LG_n^C)$,

$$0 \le \sum_{y\in N_{LG_n}(x)} f(y) \le 1.$$

Consequently,

$$\sum_{y\in N_{LG_n^C}[x]} f(y) = 2 - \sum_{y\in N_{LG_n}(x)} f(y) \ge 1.$$

Thus the first condition in the definition of an SRDF is satisfied.

We now verify the second condition. The vertices assigned value -1 are

$$(1,k), \quad 2 \le k \le n-1,$$

together with

$$(2,2) \quad \text{and} \quad (2,n-1).$$

The only vertices assigned value 2 are $(1,1)$ and $(1,n)$. If $v=(1,k)$ with $2 \le k \le n-1$, then v is adjacent in LG_n only to $(1,k-1)$, $(1,k+1)$, and $(2,k)$. Hence v is adjacent in LG_n^C to at least one of the two vertices $(1,1)$ and $(1,n)$. Indeed, when $k=2$, the vertex $(1,2)$ is not adjacent to $(1,n)$ in LG_n, and so it is adjacent to $(1,n)$ in LG_n^C. Similarly, when $k=n-1$, the vertex $(1,n-1)$ is adjacent to $(1,1)$ in LG_n^C. For $3 \le k \le n-2$, the vertex $(1,k)$ is adjacent in LG_n^C to both $(1,1)$ and $(1,n)$.

Moreover, the vertex $(2,2)$ is not adjacent to $(1,1)$ in LG_n, and hence $(2,2)$ is adjacent to $(1,1)$ in LG_n^C. Similarly, $(2,n-1)$ is not adjacent to $(1,n)$ in LG_n, and hence $(2,n-1)$ is adjacent to $(1,n)$ in LG_n^C. Therefore every vertex labelled -1 is adjacent in LG_n^C to a vertex labelled 2. Thus the second condition is also satisfied.

Hence f is a SRDF on LG_n^C of weight 2. Therefore

$$\gamma_{SR}(LG_n^C) \le 2.$$

It remains to show that no SRDF of smaller weight exists.

Let

$$g : V(LG_n^c) \longrightarrow \{-1,1,2\}$$

be an arbitrary SRDF on LG_n^c, and let

$$W = \omega(g) = \sum_{v \in V(LG_n^c)} g(v).$$

We prove that $W \ge 2$.

For a vertex $x \in V(LG_n)$, write

$$T(x) = \sum_{y \in N_{LG_n}(x)} g(y),$$

where $N_{LG_n}(x)$ is the open neighbourhood of x in the original ladder graph LG_n. Since LG_n^c is the complement of LG_n, we have,

$$N_{LG_n^c}[x] = V(LG_n) \setminus N_{LG_n}(x).$$

Therefore,

$$\sum_{y \in N_{LG_n^c}[x]} g(y) = W - T(x).$$

The first condition for an SRDF on LG_n^c gives

$$W - T(x) \ge 1.$$

Equivalently,

$$T(x) \le W - 1, \qquad \text{for every } x \in V(LG_n). \tag{3.1}$$

Suppose, for contradiction, that $W \le 1$. We split into two cases.

Case 1: $W \le 0$.

Summing inequality (3.1) over all vertices $x \in V(LG_n)$, we obtain

$$\sum_{x \in V(LG_n)} T(x) \le 2n(W-1). \tag{3.2}$$

On the other hand, every vertex $v \in V(LG_n)$ is counted in this sum exactly $\deg_{LG_n}(v)$ times. The ladder graph LG_n has four vertices of degree 2 and all remaining vertices have degree 3. Let C denote the set of the four corner vertices of LG_n. Then

$$\sum_{x \in V(LG_n)} T(x) = \sum_{v \in V(LG_n)} \deg_{LG_n}(v) g(v) = 3W - \sum_{v \in C} g(v). \tag{3.3}$$

Combining (3.2) and (3.3), we get

$$3W - \sum_{v \in C} g(v) \leq 2n(W-1).$$

Hence

$$2n \leq (2n-3)W + \sum_{v \in C} g(v). \tag{3.4}$$

Since $W \leq 0$ and $g(v) \leq 2$ for every vertex v, we have

$$(2n-3)W + \sum_{v \in C} g(v) \leq \sum_{v \in C} g(v) \leq 8.$$

Thus (3.4) gives

$$2n \leq 8.$$

For $n \geq 5$, this is impossible.

It remains only to consider $n = 4$. In this case (3.4) forces equality throughout. Hence $W = 0$ and

$$\sum_{v \in C} g(v) = 8.$$

Therefore all four corner vertices must have label 2. But then the four non-corner vertices have total weight

$$W - 8 = -8,$$

which is impossible because each vertex has label at least -1, so four vertices have total weight at least -4. Thus we get a contradiction in this case.

Case 2: $W = 1$.

Then inequality (3.1) becomes

$$T(x) \leq 0 \qquad \text{for every } x \in V(LG_n). \tag{3.5}$$

Again summing over all vertices and using (3.3), we get

$$\sum_{x \in V(LG_n)} T(x) = 3W - \sum_{v \in C} g(v) = 3 - \sum_{v \in C} g(v). \tag{3.6}$$

Since every $T(x) \leq 0$, the right-hand side must be non-positive. Hence

$$\sum_{v \in C} g(v) \geq 3.$$

We now show that no corner vertex can have label 2. Write

$$a_i = g((1,i)), \qquad b_i = g((2,i)).$$

Consider the left endpoint inequalities obtained from (3.5):

$$T((1,1)) = a_2 + b_1 \leq 0,$$

and

$$T((2,1)) = b_2 + a_1 \leq 0.$$

If $b_1 = 2$, then $a_2 + b_1 \geq -1 + 2 = 1$, contradiction. Thus $b_1 \neq 2$. Similarly, $a_1 \neq 2$.

At the right endpoint, the inequalities

$$T((1,n)) = a_{n-1} + b_n \leq 0,$$

and

$$T((2,n)) = b_{n-1} + a_n \leq 0$$

show that $b_n \neq 2$ and $a_n \neq 2$.

Therefore every corner label is either -1 or 1. We have, the sum of the four corner labels is at least 3. Since each of them is now in $\{-1, 1\}$, it follows that all four corner vertices must have label 1. Hence

$$a_1 = b_1 = a_n = b_n = 1.$$

Using the endpoint inequalities once more

$$a_2 = -1, \qquad b_2 = -1, \qquad a_{n-1} = -1, \qquad b_{n-1} = -1.$$

Now consider the vertices in the second column. From (3.5), we have

$$T((1,2)) = a_1 + a_3 + b_2 \leq 0.$$

Using $a_1 = 1$ and $b_2 = -1$, this becomes

$$a_3 \leq 0.$$

Since $a_3 \in \{-1, 1, 2\}$, we must have

$$a_3 = -1.$$

Similarly,

$$T((2,2)) = b_1 + b_3 + a_2 \leq 0$$

implies

$$b_3 = -1.$$

Consequently,

$$T((1,2)) = -1 \qquad \text{and} \qquad T((2,2)) = -1. \tag{3.7}$$

But from (3.6) and the fact that all four corner vertices have label 1, we get

$$\sum_{x \in V(LG_n)} T(x) = 3 - 4 = -1.$$

This is impossible, because (3.7) already gives two vertices whose T-values are -1, while every other $T(x) \leq 0$. Thus the total sum of all $T(x)$'s would be at most -2, contradicting the above equation, and hence we get a contradiction in this case as well.

Thus, every SRDF on LG_n^c has weight at least 2, and so

$$\gamma_{\mathrm{sR}}(LG_n^c) \geq 2.$$

Combining the upper and lower bounds gives

$$\gamma_{SR}(LG_n^C) = 2.$$

This completes the proof. □

4. Remarks on Possible Future Work

We have determined the signed Roman domination number of the ladder graph LG_n and of its complement. It would be natural to extend these computations to related graph families, including grid graphs, cylindrical grids, circular ladders, and their complements. Another direction is to study signed Roman k-domination and weak signed Roman domination for these families. Concretely, we mention the following directions.

(1) It is a natural next step to find $\gamma_{SR}(G)$ when G is either a grid graph or its complement. We leave that problem open.

(2) In a forthcoming paper, we shall look at the SRDN for the circular ladder graph and its complement.

(3) Some other closely related types of domination numbers have been studied in the literature. We point out two of these: the concept of signed Roman k-Domination in graphs by Henning and Volkmann [HV16] ($k = 1$ corresponds to a SRDF), and the concept of weak signed Roman Domination in graphs by Volkmann [Vol20]. It would be interesting to study the graphs we study in this paper for these domination numbers.

Acknowledgements

We thank the reviewer for a careful reading of our original submission and uncovering several gaps in our proofs, which improved our work and exposition substantially.

Appendix A. Finite Verification for the Ladder Lower Bound

This appendix records the finite verification used in the proof of Theorem 1. The computation involves only the nine possible row labellings

$$A = (-1, -1), B = (-1, 1), C = (-1, 2), D = (1, -1),$$
$$E = (1, 1), F = (1, 2), G = (2, -1), H = (2, 1), I = (2, 2).$$

The weight of a row labelling is the sum of its two entries.

The following code[3] evaluates the dynamic programming recurrence exactly. It also verifies the eventual recurrence

$$M_n = M_{n-2} + 1.$$

```
from itertools import product

# Row states:
# A=(-1,-1), B=(-1,1), C=(-1,2),
# D=(1,-1),  E=(1,1), F=(1,2),
# G=(2,-1),  H=(2,1), I=(2,2)

states = [
    (-1, -1),
```

[3] Generated using ChatGPT.

```
    (-1,  1),
    (-1,  2),
    ( 1, -1),
    ( 1,  1),
    ( 1,  2),
    ( 2, -1),
    ( 2,  1),
    ( 2,  2),
]

names = list("ABCDEFGHI")
state_name = dict(zip(states, names))
weight = {s: s[0] + s[1] for s in states}


def internal_admissible(P, Q, R):
    """
    Checks whether the middle row Q is valid,
    assuming its neighbouring rows are P and R.
    """
    p1, p2 = P
    q1, q2 = Q
    r1, r2 = R

    # Closed-neighbourhood sum conditions.
    if p1 + q1 + r1 + q2 < 1:
        return False
    if p2 + q2 + r2 + q1 < 1:
        return False

    # Every -1 must have a neighbour labelled 2.
    if q1 == -1 and 2 not in (p1, q2, r1):
        return False
    if q2 == -1 and 2 not in (p2, q1, r2):
        return False

    return True


def bottom_admissible(Q, R):
    """
    Checks whether the bottom row Q is valid,
    using only its vertical row-neighbour R and
    the horizontal neighbour inside Q.
    """
```

```
    q1, q2 = Q
    r1, r2 = R

    # Closed-neighbourhood sum conditions at the bottom endpoint.
    if q1 + q2 + r1 < 1:
        return False
    if q2 + q1 + r2 < 1:
        return False

    # Endpoint -1 vertices only have two neighbours.
    if q1 == -1 and 2 not in (q2, r1):
        return False
    if q2 == -1 and 2 not in (q1, r2):
        return False

    return True


def top_admissible(P, Q):
    """
    Checks whether the top row Q is valid,
    using only its vertical row-neighbour P and
    the horizontal neighbour inside Q.
    """
    p1, p2 = P
    q1, q2 = Q

    # Closed-neighbourhood sum conditions at the top endpoint.
    if p1 + q1 + q2 < 1:
        return False
    if p2 + q2 + q1 < 1:
        return False

    # Endpoint -1 vertices only have two neighbours.
    if q1 == -1 and 2 not in (p1, q2):
        return False
    if q2 == -1 and 2 not in (p2, q1):
        return False

    return True


def compute_M(n, reconstruct=False):
    """
    Computes M_n, the minimum weight of an admissible sequence
```

```
    of n row labellings.

    If reconstruct=True, also returns one minimizing sequence.
    """
    if n < 2:
        raise ValueError("This code is intended for n >= 2.")

    INF = 10**9

    # D[(P,Q)] is the minimum weight of a partial sequence
    # ending in rows P,Q, with all rows up to the penultimate
    # row already checked.
    D = {}
    back = {}

    # Initial condition: rows 1 and 2.
    for P, Q in product(states, repeat=2):
        if bottom_admissible(P, Q):
            D[(P, Q)] = weight[P] + weight[Q]
        else:
            D[(P, Q)] = INF

    # Dynamic programming transition.
    for i in range(2, n):
        new_D = {(Q, R): INF for Q, R in product(states, repeat=2)}

        for P, Q, R in product(states, repeat=3):
            if D[(P, Q)] < INF and internal_admissible(P, Q, R):
                candidate = D[(P, Q)] + weight[R]

                if candidate < new_D[(Q, R)]:
                    new_D[(Q, R)] = candidate
                    back[(i + 1, Q, R)] = P

        D = new_D

    # Apply the top endpoint condition.
    best = INF
    best_pair = None

    for P, Q in product(states, repeat=2):
        if top_admissible(P, Q) and D[(P, Q)] < best:
            best = D[(P, Q)]
            best_pair = (P, Q)
```

```
    if not reconstruct:
        return best

    # Reconstruct one minimizing sequence.
    sequence = [None] * n
    sequence[-2], sequence[-1] = best_pair

    for i in range(n, 2, -1):
        sequence[i - 3] = back[(i, sequence[i - 2], sequence[i - 1])]

    return best, sequence


def print_verification(N=30):
    print("State table:")
    for s in states:
        print(f"{state_name[s]} = {s}, weight = {weight[s]}")

    print("\nValues of M_n:")
    for n in range(2, N + 1):
        M, seq = compute_M(n, reconstruct=True)
        target = n // 2 + 2
        word = "".join(state_name[s] for s in seq)
        weights = [weight[s] for s in seq]

        print(
            f"n={n:2d}: M_n={M:2d}, "
            f"floor(n/2)+2={target:2d}, "
            f"sequence={word}, weights={weights}"
        )

    print("\nChecking formula M_n = floor(n/2)+2:")
    ok_formula = all(compute_M(n) == n // 2 + 2 for n in range(2, N + 1))
    print("Formula holds up to N =", N, ":", ok_formula)

    print("\nChecking recurrence M_n = M_{n-2}+1 for n >= 6:")
    ok_recurrence = all(
        compute_M(n) == compute_M(n - 2) + 1
        for n in range(6, N + 1)
    )
    print("Recurrence holds up to N =", N, ":", ok_recurrence)


if __name__ == "__main__":
    print_verification(30)
```

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used ChatGPT in order to generate the code which is given in Appendix A. After using this tool/service, the authors reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

[AAHL+14] H. Abdollahzadeh Ahangar, Michael A. Henning, Christian Löwenstein, Yancai Zhao, and Vladimir Samodivkin. Signed Roman domination in graphs. *J. Comb. Optim.*, 27(2):241–255, 2014.

[Ber73] Claude Berge. *Graphs and hypergraphs. Translated by Edward Minieka*, volume 6 of *North-Holland Math. Libr.* Elsevier (North-Holland), Amsterdam, 1973.

[BVA16] A. Behtoei, E. Vatandoost, and Rajol Abad F. Azizi. Signed Roman domination number and join of graphs. *J. Algebr. Syst.*, 4(1):65–77, 2016.

[GJ79] Michael R. Garey and David S. Johnson. Computers and intractability. A guide to the theory of NP-completeness. A Series of Books in the mathematical Sciences. San Francisco: W. H. Freeman and Company. X, 338 p. (1979)., 1979.

[HV16] Michael A. Henning and Lutz Volkmann. Signed Roman k-domination in graphs. *Graphs Comb.*, 32(1):175–190, 2016.

[HYZZ20] Xia Hong, Tianhu Yu, Zhengbang Zha, and Huihui Zhang. The signed Roman domination number of two classes graphs. *Discrete Math. Algorithms Appl.*, 12(2):18, 2020. Id/No 2050024.

[Ore62] Øystein Ore. *Theory of graphs*, volume 38 of *Colloq. Publ., Am. Math. Soc.* American Mathematical Society (AMS), Providence, RI, 1962.

[Vol20] Lutz Volkmann. Weak signed Roman domination in graphs. *Commun. Comb. Optim.*, 5(2):111–123, 2020.

University of Duhok, University campus, Zakho street, Duhok, Kurdistan region, Iraq

Email address: dilpakhaje@uod.ac, delbrin.ahmed@uod.ac

School of Mathematics, University of Leeds, Leeds, LS2 9JT, UK

Email address: h.izanloo@leeds.ac.uk

Mathematical and Physical Sciences division, School of Arts and Sciences, Ahmedabad University, Navrangpura, Ahmedabad - 380009, Gujarat, India

Email address: manjil@saikia.in