Soft Comput (2018) 22:6583–6593
https://doi.org/10.1007/s00500-017-2706-4

Improved integer linear programming formulation for weak Roman domination problem

Marija Ivanović[1]

Published online: 10 July 2017

Abstract Let $f : V \to \{0, 1, 2\}$ be a function, $G = (V, E)$ be a graph with a vertex set V and a set of edges E and let the weight of the vertex $u \in V$ be defined by $f(u)$. A vertex u with property $f(u) = 0$ is considered to be defended with respect to the function f if it is adjacent to a vertex with positive weight. Further, the function f is called a weak Roman dominating function (WRDF) if for every vertex u with property $f(u) = 0$ there exists at least one adjacent vertex v with positive weight such that the function $f' : V \to \{0, 1, 2\}$ defined by $f'(u) = 1$, $f'(v) = f(v) - 1$ and $f'(w) = f(w)$, $w \in V \setminus \{u, v\}$ has no undefended vertices. In this paper, an optimization problem of finding the WRDF f such that $\sum_{u \in V} f(u)$ is minimal, known as the weak Roman domination problem (WRDP), is considered. Therefore, a new integer linear programing (ILP) formulation is proposed and compared with the one known from the literature. Comparison between the new and the existing formulation is made through computational experiments on a grid, planar, net and randomly generated graphs known from the literature and up to 600 vertices. Tests were run using standard CPLEX and Gurobi optimization solvers. The obtained results demonstrate that the proposed new ILP formulation clearly outperforms the existing formulation in the sense of solutions' quality and running times.

Keywords Weak Roman domination in graphs · Combinatorial optimization · Integer linear programming

Communicated by V. Loia.

This research was partially supported by the Serbian Ministry of Education, Science and Technological Developments under the Grant No. TR36006.

Electronic supplementary material The online version of this article (doi:10.1007/s00500-017-2706-4) contains supplementary material, which is available to authorized users.

✉ Marija Ivanović
maria.ivanovic@gmail.com; marijai@math.rs

[1] Faculty of Mathematics, Univeristy of Belgrade, Studentski Trg 16, Belgrade, Serbia

1 Introduction

Problem of domination is a very popular area in a graph theory. Inspired by the real-world historical problem, in his article "Defend the Roman Empire!" Stewart (1999), Ian Stewart introduced a Roman domination problem (RDP) as a problem of organizing Roman legions such that any province without legion stationed within it must be adjacent to a province with at least two legions stationed within it. Hence, in case of an attack, when two legions are stationed within one province, one legion is considered to be permanently stationed in order to keep that province safe, while the second legion could move in order to defend an adjacent province.

In order to keep the Roman Empire safe against one attack, several generalizations and variations followed. This paper is devoted to a one of them, known as weak Roman domination problem.

Weak Roman domination problem (WRDP) was proposed by Henning and Hedetniemi (2003) as an alternative approach for defending the Empire against a single attack and can be interpreted as follows. Given that every province without legion stationed within it is vulnerable to an attack, the WRDP requires that such a province must be adjacent to a province with at least one legion stationed within it, i.e., province is considered to be defended if there is a legion stationed within it or stationed within the adjacent province.

Furthermore, movement of a legion to an adjacent province which is without any legion stationed within it must not create an undefended province.

In a graph terminology, for a graph $G = (V, E)$ and a function f, $f : V \to \{0, 1, 2\}$, a vertex $u \in V$ with property $f(u) = 0$ is considered to be defended, with respect to f, if it is adjacent to at least one vertex $v \in N_u = \{v|(u, v) \in E\}$ with positive weight ($f(v) > 0$). Further, the function f is called a weak Roman dominating function (WRDF) for a graph G if for every vertex $u \in V$ such that $f(u) = 0$ there exists an adjacent vertex $v \in V$ such that $f(v) > 0$ and a function $f' : V \to \{0, 1, 2\}$, defined by $f'(u) = 1$, $f'(v) = f(v) - 1$ and $f'(w) = f(w)$, $w \in V \setminus \{u, v\}$ has no undefended vertices. The weight of the function f is calculated by a formula $w(f) = \sum_{u \in V} f(u)$ and the minimal weight among all WRDF f for a graph G is denoted by $\gamma_r(G)$. The WRDP can be described as a problem of finding the WRDF f for a graph G with the minimal weight.

However, the WRDP is not applicable to a war strategic problems only, and it can be also used for a huge number of recourse sharing problems. For instance, emergency vehicles are very expensive and minimizing their numbers, while still being able to help en injured people, will be of great use. Therefore, if emergency vehicles are organized between hospitals as Roman legions between their provinces, all hospitals will either own a vehicle or there will be en emergency vehicle in a neighborhood hospital that can be borrowed.

This paper is organized as follows. A previous work is given in Sect. 2, while the new improved integer linear programing formulation is presented in Sect. 3. Computational results are summarized in Sect. 4.

2 Previous work

Let $G = (V, E)$ be a simple undirected graph with a vertex set V and a set of edges E. A subset $V' \subseteq V$ is called dominating in a graph G, if every vertex in $V \setminus V'$ is adjacent to some vertex in V'. The cardinality of the minimum dominating set in G, denoted by $\gamma(G)$, is called the domination number of a graph G, and a problem of finding the set V' is known as the domination problem.

The function f, $f : V \to \{0, 1, 2\}$ defined such that every vertex $u \in V$ with property $f(u) = 0$ is adjacent to a vertex $v \in V$ with property $f(v) = 2$ is called a Roman dominating function (RDF). The weight of the function f is calculated by a formula $w(f) = \sum_{v \in V} f(v)$, and a problem of finding the RDF f with the minimal weight is called a Roman domination problem (RDP). The minimal weight among all RDF represents a Roman domination number, denoted by $\gamma_R(G)$.

The RDP was initially proposed by Stewart (1999) and ReVelle and Rosing (2000). Later, with the potential of saving the Empire substantial costs of maintaining legions while still defending the Roman Empire, several generalizations and improvements arose. Some of them are the Roman k-domination Henning (2003), the weak Roman domination Henning and Hedetniemi (2003), the restrained Roman domination Pushpam and Mai (2011), the edge Roman domination Chang et al. (2014), the secure Roman domination Burger et al. (2013), the total Roman domination Liedloff et al. (2005), the independent Roman domination Targhi et al. (2012), the signed Roman domination Ahangar et al. (2014) and the strong Roman domination Alvarez-Ruiz et al. (2015).

In this paper, the weak Roman domination problem (WRDP) is considered, and therefore, the previous work will be related only to that topic.

First, Henning and Hedetniemi (2003) observed that every RDF in a graph G is also a WRDF in a G and proved that for every graph G, $\gamma(G) \leq \gamma_r(G) \leq 2\gamma(G)$. Then, they proved that $\gamma_r(P_n) = \lceil 3n/7 \rceil$ for a path P_n, $n \geq 1$, noting that cost savings of the weak Roman domination over the Roman domination number for a mentioned path, $\gamma_R(P_n) - \gamma_r(P_n) = \lceil 2n/7 \rceil - \lceil 3n/7 \rceil$, is either $\lfloor 5n/21 \rfloor$ or $\lceil 5n/21 \rceil$ and showed that $\gamma_r(C_n) = \gamma_r(P_n)$ for cycles C_n when $n \geq 4$. And finally they characterized graphs G for which $\gamma_r(G) = \gamma(G)$ and forests G for which $\gamma_r(G) = 2\gamma(G)$.

Considering properties of domination, Roman domination, weak Roman domination and secure domination problems, Cockayne et al. (2005) studied four parameters which give the minimum number of legions required to protect the graph under the mentioned strategies and obtained the exact values for specific classes of graphs. Also, they gave characterization of secure dominating sets which are minimal. Setting $\gamma_s(G)$ as the secure domination number on a graph G, Cockayne et. al. first proved that inequalities

$$\gamma(G) \leq \gamma_r(G) \leq \begin{cases} \gamma_R(G) \leq 2\gamma(G) \\ \gamma_s(G) \end{cases}$$

hold. Then, for a complete graph K_n, they gave proposition that $\gamma(G) = \gamma_r(G) = \gamma_s(G) = 1$, while $\gamma_R(G) = 2$ and for a complete bipartite graph $K_{p,q}$, where $p \leq q$, they found values for $\gamma(G)$, $\gamma_R(G)$, $\gamma_r(G)$ and $\gamma_s(G)$ proving that

$$\gamma_r(G) = \begin{cases} 2, \ p = 1, 2, q > 1 \\ 3, \ p = 3 \\ 4, \ p \leq 4. \end{cases}$$

Finally, they proposed all four domination numbers for a complete multipartite graph $K_{p_1,p_2,\ldots,p_t}$ where $p_1 \leq p_2 \leq \ldots \leq p_t$ and $t \geq 3$ (proving that γ_r is equal to 2 when $p_1 = 1, 2$ and equal to 3 when $p_1 \geq 3$), paths P_n and cycles C_n (proving that $\gamma_r = \gamma_s = \lceil 3n/7 \rceil$), and their products $P_m \times P_k$ and $C_m \times C_k$ (proving that $\gamma_r(P_m \times P_k) \leq \gamma_s(P_m \times P_k) \leq \lceil mk/3 \rceil + 2$ and $\gamma_r(C_m \times C_k) \leq \gamma_s(C_m \times C_k) \leq \lceil mk/3 \rceil$).

Pushpam and Mai (2011) characterized the class of the trees and split graphs for which $\gamma_r(G) = \gamma(G)$, found γ_r value for caterpillar, $2 \times n$ grid graphs and a complete binary tree and proved that $\gamma_r(C_n) = \gamma_r(P_n)$.

The weak Roman domination number of a corona of any two graphs and generalized web graphs is established by Lai et al. (2011). For instance, they prove that for a corona of a graph G of n vertices and a complete graph K_m, $\gamma_r(G \circ K_m) = n$, while for two graphs G and H with n and m vertices and $H \neq K_m$, $\gamma_r(G \circ H) = 2n$. Further, for a helm graph H_n and a web graph W_n, they show that $\gamma_r(H_n) = \gamma_r(W_n) = n + 1$, while for a generalized web graph $W_{t,3}$, when $t \geq 3$, $\gamma_r(W_{t,3}) = t + 2$. Proving that $\gamma_r(W_{3,4}) = 7$ and $\gamma_r(W_{3,5}) = 9$, they also gave generalization of a weak Roman domination number for a generalized web graphs $\gamma_r(W_{3,n})$ when $n \geq 6$ by setting $n = 3a + b$, for $0 \leq b \leq 2$ and proving that $\gamma_r(W_{3,n}) = 4a + 2$ when $b = 0$ and $\gamma_r(W_{3,n}) = 4a + 3 + b$ otherwise.

The impact on the weak Roman domination number of a simple connected graph G with connectivity of one, when one of its cut points is removed, is, for example, considered by Song et al. (2013).

The weak Roman domination number for $2 \times n$ grid graphs is determined by Song et al. (2011).

Nonetheless, some relations between several different domination numbers were nicely summarized by Chellali et al. (2014) who also showed that for all graphs, the weak Roman domination number is bounded above by the 2-rainbow domination number $\gamma_{r2}(G)$.

Motivated by Chellali et al. (2014), Alvarez-Ruiz et al. (2015) provided a constructive characterization of the trees for which the Roman domination number strongly equals to the weak Roman domination number. With the characterization based on five simple extension operations, Alvarado et al. (2015b) also revealed several structural properties of these trees.

Later, Alvarado et al. (2015a) proved that $\gamma_{r2}(G) \leq 2\gamma_r(G)$ for every graph G. Then, for a complete graph K_n they characterized the extremal graphs for this inequality that are $\{K_4, K_4 - e\}$-free, and showed that the recognition of the K_5-free extremal graphs is NP-hard (by the $K_n - e$ it was meant on a graph that arises by removing one edge from K_n).

Recently, Pushpam and Kamalam (2015a) found an efficient weak Roman domination number of some of the Myscielski graphs.

The idea of efficiency to the weak roman domination was extended by Pushpam et al, who also characterized certain classes of graphs that are efficiently weak Roman dominant (Pushpam and Kamalam 2015b).

In general case, it was proved that the RDP is NP-complete (i.e., Dreyer 2000; Shang and Hu 2007; Klobučar and Puljić 2014), but for some special classes of graphs, such as interval graphs, intersection graphs, co-graphs and distance-hereditary graphs, it can be solved in a linear time, see Klobučar and Puljić (2014). Given that the WRDP is a generalization of the RDP, the WRDP can be also considered as NP-complete problem. Nonetheless, Henning and Hedetniemi (2003) proved that WRDP is NP-complete, even restricted to bipartite and chordal graphs.

The linear time algorithm for computing the weak Roman domination number of a block graph is given by Liu et al. (2010).

Considering the algorithms for solving the WRDP, Chapelle et al. (2013) broke the trivial enumeration barrier of $O^*(3^n)$ (the notation $O^*(f(n))$ suppresses polynomial factors) providing two faster algorithms. Proving that the WRDP can be solved in $O^*(2^n)$ time needing exponential space, they described an $O^*(2.2279^n)$ algorithm using polynomial space. Their proposed results rely on structural properties of a WRDF, as well as on the best polynomial space algorithm for the Red-Blue Dominating Set problem.

Burger et al. Burger et al. (2013) gave relation between $\gamma(G)$, $\gamma_R(G)$ and $\gamma_r(G)$ and proposed the only one integer linear programming (ILP) formulation for the weak Roman domination problem to the knowledge of the author.

In this paper, a new ILP formulation will be proposed, proved to be correct and compared with the existing one. Thus, the first binary programing formulation of the WRDP will be reviewed here below.

For a graph $G = (V, E)$ represented by a vertex set V, $n = |V|$, and a set of edges E, $m = |E|$, let the adjacency matrix be denoted by a_{ij} for all $i \neq j$, with the convention that $a_{ii} = 1$ for all $i = 1, \ldots, n$. Further, let X represents a set of vertices containing exactly one legion and let Y represents a set of vertices containing exactly two legions within it. Binary decision variables will be defined with

$$x_i = \begin{cases} 1, & i \in X \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

$$y_i = \begin{cases} 1, & i \in Y \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

for all $i = 1, \ldots, n$ and

$$z_{ij} = \begin{cases} 1, & i\ (x_i = 0 \text{ and } y_i = 0) \text{ and } j\ (x_j = 1 \text{ or } y_j = 1) \\ & \text{form a swap set} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

for all $i, j = 1, \ldots, n, i \neq j$ and $i, j \in V$.

Burger et. al. formulation of the weak Roman domination problem is:

$$\min \sum_{k=1}^{n} (x_k + 2y_k) \tag{4}$$

subject to the constraints

$$\sum_{j=1}^{n} a_{ij}(x_j + y_j) \geq 1, \quad i = 1, \ldots, n \tag{5}$$

$$x_k + y_k \leq 1, \quad k = 1, \ldots, n \tag{6}$$

$$x_i + y_i + \sum_{\substack{j=1 \\ j \neq i}}^{n} z_{ij} \geq 1 \quad i = 1, \ldots, n \tag{7}$$

$$a_{ij}(x_j + y_j - x_i - y_i) \geq 2z_{ij}, \quad i, j = 1, \ldots, n, j \neq i \tag{8}$$

$$y_j a_{kj} + a_{ki} + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}}^{n} a_{kl}(x_l + y_l) \geq z_{ij},$$

$$i, j, k = 1, \ldots, n, i \neq j \tag{9}$$

$$x_i, y_i, z_{ij} \in \{0, 1\}, \quad i, j = 1, \ldots, n. \tag{10}$$

The objective function value, given by (4), represents the weak Roman domination number $\gamma_r(G)$. Constraints (5) ensure that each vertex $i \in V$ with property $f(i) = 0$ is adjacent to at least one vertex j with property $f(j) > 0$. By the constraints (6), it is ensured that sets X and Y are mutually disjunctive. Further, by the constraints (7) it is ensured that for each unoccupied vertex i ($x_i = y_i = 0$) there exists a vertex j, $j \in N_i$ with whom it can form a swap, while by the constraints (8) it is ensured that each swap from j to i is valid. Furthermore, safety of every province after any single swap performed from j to i is preserved by the constraints (9). Constraints (10) deal with binary nature of decision variables x_i, y_i and z_{ij}.

Presented formulation has $n^2 + 2n$ variables, has $n^3 + n^2 + 3n$ constraints and will be referred as the *Old*.
Note that in Burger et al. (2013), in the constraints (9) originally stand

$$y_j a_{ki} + a_{kj} + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}}^{n} a_{kl}(x_l + y_l) \geq z_{ij}.$$

We assume that it was a typing error since in that case mathematical formulation does not correspond to the WRDP.

3 Improved ILP formulation for the weak Roman domination problem

Inspired by the *Old* formulation, it was noticed that there is no need to calculate swap z_{ij} for each pair of vertices i and j. Instead of that, it is sufficient to calculate swap only between two adjacent vertices. Therefore, let $E' = \{(j, i) | (i, j) \in E\}$ and

$$z_e = \begin{cases} 1, & e = (i, j), \\ & i\ (x_i = 0 \text{ and } y_i = 0) \text{ and } j\ (x_j = 1 \text{ or } y_j = 1) \\ & \text{form a swap set} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

for all $e \in E \cup E'$ and, again, let binary decision variables x_i and y_i be defined with (1) and (2).

Now, new ILP formulation for the WRDP, on a graph $G = (V, E)$, can be described as follows:

$$\min \sum_{k \in V} (x_k + 2y_k) \tag{12}$$

subject to the constraints

$$x_i + y_i + \sum_{e \ni i} z_e \geq 1, \quad i \in V \tag{13}$$

$$x_j + y_j - x_i - y_i + 1 \geq 2z_e, \quad e = (i, j) \in E \cup E' \tag{14}$$

$$y_j + \sum_{\substack{l \in N_k \\ l \neq i, \\ l \neq j}} (x_l + y_l) \geq z_e,$$

$$e = (i, j) \in E \cup E', k \in N_j \setminus N_i \tag{15}$$

$$x_i, y_i, z_e \in \{0, 1\} \quad i \in V, e \in E \cup E'. \tag{16}$$

Again, the objective function value, given by (12), represents the weak Roman domination number $\gamma_r(G)$. Constraints (13) ensure that vertex i is defended, or it is adjacent to a defended vertex with whom it can form a swap. Validity of the swap from the vertex j to the vertex i is ensured by the constraints (14). Also, by the constraints (15), safety of the Empire is preserved after any single swap performed from j to i. Binary nature of decision variables is ensured by (16).

New ILP formulation will be referred as the *New* and has $2n + m$ variables and $3n + 2m$ constraints.

Note that, for a complete planar graph with n vertices, the *New* formulation consists of $\frac{1}{2}n^2 + \frac{3}{2}n$ variables and $n^2 + 2n$ constraints. Therefore, when the *New* formulation is used, the number of the variables is reduced by at least $\frac{1}{2}(n^2 + n)$ while the number of the constraints is reduced by $n^3 + n$.

The following theorem proves that *New* formulation is equivalent to the *Old* formulation.

Theorem 1 *For every simple graph G, optimal objective function value of the Old formulation (4)–(10) is equal to the optimal objective function value of the New formulation (12)–(16).*

Proof ($\Rightarrow$) Let a feasible solution to the *Old* formulation be represented by a set (X', Y', Z'), $X' = (x'_1, \ldots, x'_n)$, $Y' = (y'_1, \ldots, y'_n)$, $Z' = (z'_{11}, \ldots, z_{nn})$, $n = |V|$, and similarly,

let a set (X'', Y'', Z'') of variables x_i'', y_i'' and z_e'' be defined such that $x_i'' = x_i'$, $y_i'' = y_i'$, $i = 1, \ldots, n$ and

$$z_e'' = \begin{cases} 1, & e = (i, j), z_{ij}' = 1 \\ 0, & e = (i, j), z_{ij}' = 0 \end{cases}, \quad e \in E \cup E'. \tag{17}$$

Note that variables z_{ij}' are defined for each pair of vertices (i, j), while z_e'' are defined only for edges that exist in a graph G. Hence, z_e'' are defined only when $a_{ij} = 1, e = (i, j) \in E \cup E'$. Also note that, by the conditions (8), for $a_{ij} = 0$ it follows that $z_{ij}' = 0$. Thus, variables z_e'' are well defined since in the conditions (7) $z_{ij}' = 0$ are part of the sum, and given that zero is neutral for summing, they do not affect the given sum. Even more, for those z_{ij}', conditions (8) and (9) become trivial. From the above consideration, for each $i \in V$, it follows that $\sum_{j \in V} z_{ij}'$ is equal to the $\sum_{e \ni i} z_e''$. Therefore, conditions (13) follow from the conditions (7). Also note that conditions (8) are nontrivial only when $a_{ij} = 1$. Given that, and from the definition of the z_e'', directly follows that conditions (7) imply (14).

In order to prove conditions (15), it is necessary to consider next three cases:

Case 1 $k \notin N_j$. Since k and j are not in the neighborhood, it follows that $a_{kj} = 0$, which means that left-hand side (LHS) of the conditions (9),

$$y_j' a_{kj} + a_{ki} + \sum_{l=1}^{n} a_{kl}(x_l' + y_l') - a_{ki}(x_i' + y_i') - a_{kj}(x_j' + y_j')$$

becomes

$$-a_{kj}x_j' + (1 - x_i' - y_i')a_{ki} + \sum_{l=1}^{n} a_{kl}(x_l' + y_l').$$

Further, $a_{kj}x_j' = 0$ holds from the starting assumption, while from the conditions (6) it follows that $1 - x_i' - y_i' \geq 0$. Finally, from the conditions (5) it follows that

$$-a_{kj}x_j' + (1 - x_i' - y_i')a_{ki} + \sum_{l=1}^{n} a_{kl}(x_l' + y_l') \geq \sum_{l=1}^{n} a_{kl}(x_l' + y_l') \geq 1 \geq z_{ij}'.$$

From the above consideration, it is obvious that in *Case 1* conditions (9) are redundant.

Case 2 $k \in N_i \cap N_j$. By the assumption that the vertex k is in the neighborhood of the vertices i and j, it follows that $a_{ki} = a_{kj} = 1$. Since $a_{kj}y_j' \geq 0$ and $\sum_{\substack{l=1 \\ l \neq i \\ l \neq j}} a_{kl}(x_l' + y_l') \geq 0$, it follows that LHS of the conditions (9) is greater or equal to $a_{ki} = 1$, implying that LHS is greater or equal to z_{ij}'.

From the above consideration, it is obvious that in *Case 2* conditions (9) are redundant.

Case 3 $k \in N_j \setminus N_i$. Assuming that vertex k is in the neighborhood of the j and not in the neighborhood of the i, it follows that $a_{kj} = 1$ and $a_{ki} = 0$. Therefore, conditions (9) becomes $y_j' + \sum_{\substack{l=1 \\ l \neq i \\ l \neq j}} a_{kl}(x_l' + y_l') \geq z_{ij}'$. By the definition of the variables z_e'', it follows that conditions (15) hold.

Since that *Case 1* and *Case 2* are redundant and do not have influence on the feasible solution space of the model (4)–(10), it follows that they can be ignored.

Variables x_i'', y_i'' and z_e'' are binary by the definition, i.e., conditions (16) hold also.

Finally, since $x_k'' = x_k'$ and $y_k'' = y_k'$, it follows that objective function values are equal, i.e.,

$$Obj_{Old} = \sum_{k \in V} (x_k' + 2y_k') = \sum_{k \in V} (x_k'' + 2y_k'') = Obj_{New}.$$

Even more, objective function value of the *Old* formulation is greater or equal to the objective function value of the *New* formulation, since feasible solution of the *Old* formulation is feasible solution of the *New* formulation.

($\Leftarrow$) Let a set (X'', Y'', Z'') represents a feasible solution to the *New* formulation $(X'' = (x_1'', \ldots, x_n''), Y'' = (y_1'', \ldots, y_n''), Z'' = (z_1'', \ldots, z_m''))$, $n = |V|, m = |E|$ and let a set (X', Y', Z') of variables x_i', y_i' and z_{ij}', $i, j = 1, \ldots, n$ be defined such that $y_i' = y_i''$, $i = 1, \ldots, n$,

$$x_i' = \begin{cases} x_i'', & x_i'' + y_i'' \leq 1 \\ x_i'' - 1, & x_i'' + y_i'' = 2 \end{cases}$$

and

$$z_{ij}' = \begin{cases} z_e'', & x_j'' + y_j'' \leq 1, e = (i, j) \in E \cup E' \\ 0, & \text{otherwise.} \end{cases}$$

In order to prove conditions (5)–(9), two cases will be observed:

(*1*) $x_i'' + y_i'' \leq 1$ and
(*2*) $x_i'' + y_i'' = 2$.

Conditions (6) trivially hold since, in (*1.*), $x_k' + y_k' = x_k'' + y_k'' \leq 1$ and, in (*2.*), $x_k' + y_k' = x_k'' - 1 + y_k'' = 1 \leq 1$.

Starting from the conditions (13) and by the definition of the variables z_{ij}', conditions (7) hold also:

$$x'_i + y'_i + \sum_{\substack{j=1\\ j\neq i}}^{n} z'_{ij} =$$

in (1.), $x''_i + y''_i +$

$$\sum_{\substack{e=(i,j)\in E\cup E'\\ x''_j+y''_j\leq 1}} z'_{ij} + \sum_{\substack{e=(i,j)\in E\cup E'\\ x''_j+y''_j=2}} z'_{ij} + \sum_{e=(i,j)\notin E\cup E'} z'_{ij} =$$

$$x''_i + y''_i + \sum_{\substack{e=(i,j)\in E\cup E'\\ x''_j+y''_j\leq 1}} z'_{ij} =$$

$$x''_i + y''_i + \sum_{e\ni i} z''_e - \sum_{\substack{e=(j,i)\in E\cup E'\\ x''_j+y''_j=2}} z''_e =$$

$$x''_i + y''_i + \sum_{e\ni i} z''_e \geq 1$$

in (2.), $x''_i - 1 + y''_i + \sum_{\substack{j=1\\ j\neq i}}^{n} z'_{ij} \geq x''_i - 1 + y''_i = 1$

for all $i = 1, \ldots, n$.

Above formulas are correct since $\sum_{\substack{e=(i,j)\in E\cup E'\\ x''_j+y''_j=2}} z'_{ij} = 0$ and $\sum_{e=(i,j)\notin E\cup E'} z'_{ij} = 0$ by the definition of the variable z'_{ij}, while $\sum_{\substack{e=(j,i)\in E\cup E'\\ x''_j+y''_j=2}} z''_e = 0$ because of the constraint (14) which in case $e = (j, i)$ should read as follows $x''_i + y''_i - x''_j - y''_j + 1 \geq 2z''_e$ directly implying that $z''_e = 0$.

Validity of the conditions (8) can be shown also: For $a_{ij} = 0$, conditions (8) trivially hold since $0 \geq 0$, while for $a_{ij} = 1$ four cases are considered:

– (*1.*) holds for all $i, j = 1, \ldots, n$.

$$x'_j+y'_j-(x'_i+y'_i)+1 = x''_j+y''_j-(x''_i+y''_i)+1 \geq 2z''_e = 2z'_{ij}.$$

– (*1.*) holds for i while (*2.*) holds for j, $i, j = 1, \ldots, n$.

$$x'_j + y'_j - (x'_i + y'_i) + 1 = x''_j - 1 + y''_j - (x''_i + y''_i) + 1$$
$$= x''_j + y''_j - (x''_i + y''_i) = 2 - (x''_i + y''_i) \geq 1 \geq 0 = 2z'_{ij}$$

since, by the definition, $z'_{ij} = 0$ when $x''_j + y''_j = 2$.

– (*1.*) holds for j while (*2.*) holds for i, $i, j = 1, \ldots, n$.

$$x'_j + y'_j - (x'_i + y'_i) + 1 = x''_j + y''_j - (x''_i - 1 + y''_i) + 1$$
$$= x''_j + y''_j - (x''_i + y''_i) + 2 \geq 2z''_e \geq 2z'_{ij}.$$

– (*2.*) holds for all $i, j = 1, \ldots, n$.

$$x'_j + y'_j - (x'_i + y'_i) + 1$$
$$= x''_j - 1 + y''_j - (x''_i - 1 + y''_i) + 1 =$$
$$x''_j + y''_j - (x''_i + y''_i) + 1 \geq 2z''_e \geq 2z'_{ij}.$$

Combining all four cases together with the case when $a_{ij} = 0$, it follows that conditions (8) are satisfied.

Conditions (5) which can be equally written as

$$x'_i + y'_i + \sum_{j\in N_i} (x'_j + y'_j) \geq 1$$

are satisfied for (*2.*) since

$$x'_i + y'_i + \sum_{j\in N_i} (x'_j + y'_j)$$
$$= x''_i - 1 + y''_i + \sum_{j\in N_i} (x'_j + y'_j) \geq 1 + \sum_{j\in N_i} (x'_j + y'_j) \geq 1.$$

If (*1.*) holds for i and conditions $x''_j + y''_j \leq 1$ hold for all j, conditions (5) can be written as $x''_i + y''_i + \sum_{j\in N_i} (x''_j + y''_j) \geq 1$. The last relation holds if at least one of next two relations is satisfied: $x''_i + y''_i \geq 1$ or $\sum_{j\in N_i} (x''_j + y''_j) \geq 1$. Assuming that conditions (13)–(16) are satisfied, from the conditions (13) it follows that at least one of the relations $x''_i + y''_i \geq 1$ and $\sum_{e\ni i} z''_e \geq 1$ holds. If the first relation holds, conditions (5) are satisfied. If the second relation holds, then there exists an edge $\bar{e}$ such that $z''_{\bar{e}} = 1$. Now, from the condition (14) it follows that $x''_j + y''_j \geq 1$ for $\bar{e} = (i, j)$, proving that conditions (5) are satisfied again.

If (*1.*) holds for i and there exists $\bar{j}$ such that $x''_{\bar{j}} + y''_{\bar{j}} = 2$, conditions (5) can be written as $x'_i + y'_i + \sum_{j\in N_i, j\neq\bar{j}} (x'_j + y'_j) + (x'_{\bar{j}} + y'_{\bar{j}}) \geq x'_{\bar{j}} + y'_{\bar{j}} = x''_{\bar{j}} - 1 + y''_{\bar{j}} = 1$.

In order to show validity of the conditions (9), next three cases should be considered.

1. $a_{kj} = a_{ki} = 0$.
   Keeping in mind that constraints (5) hold, constraints (9) can be equally written as

$$a_{kj} y'_j + a_{ki} + \sum_{\substack{l=1\\ l\neq i\\ l\neq j}}^{n} a_{kl} (x'_l + y'_l)$$
$$= \sum_{l=1}^{n} a_{kl} (x'_l + y'_l) - a_{ki} (x'_i + y'_i) - a_{kj} (x'_j + y'_j)$$
$$= \sum_{l=1}^{n} a_{kl} (x'_l + y'_l) \geq 1 \geq z'_{ij}$$

from which it can be concluded that they are satisfied.

2. $a_{kj} = 1, a_{ki} = 0$.
Keeping in mind that constraints (15) are also satisfied, constraints (9) are satisfied again,

$$a_{kj}y'_j + a_{ki} + \sum_{\substack{l=1\\ l\neq i\\ l\neq j}}^{n} a_{kl}(x'_l + y'_l) = y'_j + \sum_{\substack{l\in N_k\\ l\neq i\\ l\neq j}} (x'_l + y'_l) =$$

case $((\forall l)(x''_l + y''_l \leq 1))$:

$$y''_j + \sum_{\substack{l\in N_k\\ l\neq i\\ l\neq j}} (x''_l + y''_l) \geq z''_e \geq z'_{ij}$$

case $((\exists \bar{l})(x''_{\bar{l}} + y''_{\bar{l}} = 2))$:

$$y'_j + \sum_{\substack{l\in N_k\\ l\neq i\\ l\neq j\\ l\neq \bar{l}}} (x'_l + y'_l) + (x'_{\bar{l}} + y'_{\bar{l}}) \geq x'_{\bar{l}} + y'_{\bar{l}} = 1 \geq z'_{ij}$$

3. $a_{kj} = 0 \wedge a_{ki} = 1$ or $a_{kj} = a_{ki} = 1$.
Since $a_{kj}y'_i \geq 0$ and $a_{kl}(x'_l + y'_l) \geq 0$ for all $k, l = 1, \ldots, n, l \neq i$, constraints (9) are again satisfied,

$$a_{kj}y'_j + a_{ki} + \sum_{\substack{l=1\\ l\neq i\\ l\neq j}}^{n} a_{kl}(x'_l + y'_l) =$$

$$a_{kj}y'_i + 1 + \sum_{\substack{l=1\\ l\neq i\\ l\neq j}}^{n} a_{kl}(x'_l + y'_l) \geq 1 \geq z'_{ij}.$$

From the above, it can be concluded that constraints (9) hold.

Finally, since $x'_k \leq x''_k$ and $y'_k = y''_k$, it follows that objective function value of the *Old* formulation is less or equal to the objective function value of the *New* formulation, i.e., $Obj_{Old} \leq Obj_{New}$. Even more, feasible solution to the *New* formulation is a feasible solution to the *Old* formulation.

Combining given inequalities theorem is proved, i.e., $Obj_{New} = Obj_{Old}$. □

4 Computational results

In this section, computational results which show effectiveness of the proposed ILP formulation method are summarized. Direct comparison is given only between the proposed ILP formulation and mathematical formulation presented in Burger et al. (2013). Comparison with algorithms presented in Liu et al. (2010) and Chapelle et al. (2013) is not possible since both of these two papers are theoretical and without numerical results. Tests were run using CPLEX 12.6 and Gurobi 5.6 optimization solvers and carried out on Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz 2.39GHz, with 8GB RAM under Windows 8.1 operating system.

For all computational experiments grid, planar, net and randomly generated graphs were used. Grid, planar and net graphs are well-known type of graphs, while randomly generated graphs are provided by Vincenzo Currò and were specially generated for his Ph.D. theses Currò (2014). The set of grid graph instances consists of 42 instances where the smallest one has 40 vertices, while the greatest one has 100 vertices. The set of planar graph instances consists only of 6 instances with 10, 20, 30, 50, 100 and 150 vertices. The set of net graphs consists only of four instances with 100, 200, 400 and 600 vertices, while randomly generated graphs consists of 32 instances from which 18 of them are with 50 and 14 of them are with 100 vertices.

All instances were tested using both optimization solvers. The results are shown for each type of graph instance in separate tables. For all tables, in the first three columns the name of the instance, the number of vertices and the number of edges are given. In case that optimizations solvers succeeded in finding and proving the optimal solution to the tested instances, the objective function value is shown at the fourth column (*val*). For each ILP formulation, the results (objective function value and optimization solvers' execution time) are shown in the next two sets of four columns such that the results of the *Old* formulation are followed by the results of the *New* formulation. For instances where optimal solution was not reached within the time limit of 7200 seconds, the sign "*" is shown. Also for instances where optimization solver stopped because of "out of memory status," the sign "–" stands.

Comparing the execution time of two solvers and two formulations, it is obvious that both solvers were more successful by using *New* formulation.

From Table 1, based on the *Old* formulation, CPLEX reached optimal solution value for 22 instances and Gurobi for 9 instances, while based on the *New* formulation, CPLEX and Gurobi were successful in finding optimal solution for 32 instances. Actually, based on the *New* formulation, Gurobi succeeds in finding the solution value for all grid graph instances, but because of the running time limit, not all found solution values are verified as optimal. Even more, from Table 1, for all instances where solution is found and verified as optimal, Gurobi running time was less when *New* formulation is used. For instances, with at most 60 vertices CPLEX, based on the both formulations, was successful in finding optimal solution values with very small differences in run-

Table 1 Computational results on grid graph instances

Instance				*Old* formulation				*New* formulation			
	$\|V\|$	$\|E\|$	opt	val.	t_{gur}	value	t_{cpl}	val	t_{gur}	val	t_{cpl}
grid04 × 10	40	66	15	15	497.37	15	4.954	15	9.16	15	4.109
grid05 × 08	40	67	14	14	551.75	14	5.191	14	7.56	14	4.64
grid03 × 14	42	67	16	16	314.72	16	4.829	16	13.64	16	5.372
grid06 × 07	42	71	15	15	454.75	15	6.927	15	12.28	15	5.801
grid04 × 11	44	73	16	16	912.76	16	6.13	16	14.1	16	5.5
grid03 × 15	45	72	17	17	712.63	17	8.36	17	19.69	17	7.789
grid05 × 09	45	76	16	16	942.68	16	7.908	16	19.48	16	10.781
grid04 × 12	48	80	17	17	891.93	17	14.364	17	14.91	17	12.84
grid06 × 08	48	82	18	18	*	18	26.662	18	108.8	18	25.499
grid07 × 07	49	84	18	18	*	18	9.845	18	80.37	18	12.844
grid05 × 10	50	85	18	18	5953.67	18	11.469	18	71.83	18	10.61
grid04 × 13	52	87	19	22	–	19	13.16	19	69.13	19	11.813
grid06 × 09	54	93	19	55	–	19	26.988	19	79.64	19	25.539
grid05 × 11	55	94	19	59	–	19	14.365	19	107.34	19	11.424
grid04 × 14	56	94	20	25	–	20	35.6	20	52.53	20	35.326
grid07 × 08	56	97	20	23	–	20	21.882	20	107.63	20	42.48
grid04 × 15	60	101	22	27	–	22	40.256	22	172.62	22	51.518
grid05 × 12	60	103	21	28	–	21	41.364	21	127.57	21	14.88
grid06 × 10	60	104	21	26	–	21	51.458	21	58.41	21	35.713
grid07 × 09	63	110	22	27	–	22	995.68	22	101.89	22	70.259
grid08 × 08	64	112	23	26	–	23	*	23	1664.53	23	171.925
grid05 × 13	65	112	23	28	–	23	1825	23	195.49	23	67.007
grid06 × 11	66	115	24	29	–	24	*	24	1001.9	24	381.771
grid05 × 14	70	121	24	28	–	24	5368.9	24	677.63	24	73.489
grid07 × 10	70	123	25	30	–	25	*	25	829.21	25	618.089
grid06 × 12	72	126	26	31	–	27	-	26	3254.34	26	1166.405
grid08 × 09	72	127	25	32	–	25	*	25	4195.49	25	435.146
grid05 × 15	75	130	26	31	–	26	*	26	288.06	26	465.006
grid07 × 11	77	136	27	32	–	28	*	27	1543.24	27	988.596
grid06 × 13	78	137	27	33	–	28	*	27	6657.33	27	1005.126
grid08 × 10	80	142	28	33	–	28	*	28	3693.43	28	2162.812
grid09 × 09	81	144	28	35	–	29	*	28	*	28	737.579
grid06 × 14	84	148		34	–	30	*	30	*	30	–
grid07 × 12	84	149	29	35	–	30	*	29	4637.38	30	–
grid08 × 11	88	157		36	–	31	*	31	*	31	–
grid06 × 15	90	159		95	–	32	*	32	*	33	–
grid09 × 10	90	161		90	–	32	*	31	*	32	–
grid07 × 13	91	162		109	–	32	*	32	*	33	–
grid08 × 12	96	172		106	–	38	–	33	*	34	–
grid07 × 14	98	175		115	–	34	*	35	*	34	–
grid09 × 11	99	178		110	–	35	*	35	*	35	–
grid10 × 10	100	180		100	–	36	–	35	*	36	–

ning time, but for instances such as grid07 × 09, grid05 × 13 and grid05 × 14, when *New* formulation is used, running time was significantly less. Also, for instances with more than 60 vertices, where CPLEX based on the *New* formulation was successful in finding optimal solution, running time was lesser than 2200 seconds, while CPLEX based on the *Old* formulation, for the same instances, was either stopped because of "out of memory" status, either because of the time limit.

Table 2 Computational results on net graph instances

Instance				*Old* formulation				*New* formulation			
	$\|V\|$	$\|E\|$	opt	val.	t_{gur}	value	t_{cpl}	val	t_{gur}	val	t_{cpl}
net-10-10	100	342	20	90	–	21	–	20	598.98	20	148.213
net-10-20	200	712			–		–	42	*	40	–
net-20-20	400	1482			–		–	83	*	87	–
net-30-20	600	2252			–		–	122	*	142	–

Table 3 Computational results on planar graph instances

Instance				*Old* formulation				*New* formulation			
	$\|V\|$	$\|E\|$	opt	val.	t_{gur}	value	t_{cpl}	val	t_{gur}	val	t_{cpl}
plan10	10	27	3	3	0.31	3	0.156	3	0.17	3	0.206
plan20	20	105	3	3	2.75	3	1.363	3	1.36	3	1.423
plan30	30	182	5	5	26.73	5	8.724	5	11.45	5	7.49
plan50	50	465	6	6	–	6	302.82	6	98.49	6	153.215
plan100	100	1540			–		–	10	*	9	–
plan150	150	2867			–		–	13	*		–

From Table 2, it can be seen that optimal solution value for net graph instances was reached only for one instance, Net-10-10. Based on the *New* formulation, both CPLEX and Gurobi optimization solvers were stopped either because of "out of memory" status, either because of the time limitation but still were able to provide some solution values for other three instances. Based on the *Old* formulation, both optimization solvers were unsuccessful in providing any solution value for these three instances.

Table 3 consists of computational results on the planar graph instances. Optimal solution value was reached for instances with at most 50 vertices by using both optimization solvers based on the both formulations and mostly with lesser running time when the *New* formulation is used. Both optimization solvers based on the *Old* formulation were unsuccessful in solving instances plan100 and plan150 because of "out of memory" status, while Gurobi optimization solver based on the *New* formulation provides some solution value within the time limit of 2 hours.

Finally, from Table 4, computational results on randomly generated graphs are given. Usage of the *Old* formulation provides optimal solution value for 24 instances with CPLEX and 13 instances with Gurobi optimization solver, while usage of the *New* formulation provides optimal solution value for 24 instances when CPLEX is used and for 25 when Gurobi optimization solver is used. Further, for instances when optimal solution value was reached by using both formulations, at almost all instances, solvers' running times are lesser when *New* formulation is used. Moreover, on instances where provided solution value was not verified as optimal (Random-100-8, Random-100-9, Random-100-10 and Random-100-20), better solution values are provided when *New* formulation is used. Furthermore, on instances Random-100-30, Random-100-40 and Random-100-50, both optimization solvers, based on the *Old* formulation, were unsuccessful in providing any solution value before status "out of memory" occurs.

5 Conclusions

In this paper, the weak Roman domination problem is considered. New, improved ILP formulation is proposed, proved to be correct and compared with the one known from the literature. It was shown that set of the constraints (7) can be excluded, and that in some cases the set of the constraints (9), of the known formulation, are redundant. Even more, it was shown that it is sufficient to calculate swap only between two adjacent vertices. Since proposed formulation uses lesser number of constraints and lesser number of variables than the one known from the literature, significant improvements in the computational effort for solving *New* formulation were expected. Therefore, computational experiments were carried out on a grid, planar, net and randomly generated graphs up to 600 vertices from the literature. It was shown that CPLEX and Gurobi solvers, based on the *New* formulation, provide optimal solutions for a larger number of instances; then, it was the case with optimization solvers based on the *Old* formulation. Even more, running times are mostly lesser when *New* formulation is used. Regardless the fact that running times using CPLEX were lesser than the running times

Table 4 Computational results on randomly generated graph instances

Instance				*Old* formulation				*New* formulation			
	$\|V\|$	$\|E\|$	opt	val.	t_{gur}	value	t_{cpl}	val	t_{gur}	val	t_{cpl}
Random-50-1	50	49	24	28	–	24	1.125	24	0.72	24	0.281
Random-50-2	50	49	23	23	85.66	23	1.156	23	1.48	23	0.343
Random-50-3	50	58	24	24	53.95	24	1.265	24	0.66	24	0.39
Random-50-4	50	54	24	24	66.97	24	1.239	24	0.58	24	0.484
Random-50-5	50	67	22	22	79.59	22	1.635	22	1.41	22	0.968
Random-50-6	50	86	19	19	118.34	19	4.915	19	12.31	19	2.053
Random-50-7	50	84	19	19	248.14	19	4.677	19	9.33	19	3.171
Random-50-8	50	95	17	17	226.28	17	15.308	17	4.11	17	3.093
Random-50-9	50	108	17	17	865.54	17	90.188	17	39.31	17	26.373
Random-50-10	50	112	16	16	716.16	16	43.673	16	34.2	16	6.781
Random-50-20	50	248	9	13	–	9	1304.7	9	792.33	9	346.264
Random-50-30	50	373	7	9	–	7	1143.9	7	943.77	7	476.278
Random-50-40	50	475	6	9	–	6	1767.9	6	3734.27	6	1447.318
Random-50-50	50	597	5	6	–	5	1856.4	5	1545.06	5	1779.272
Random-50-60	50	739	4	6	–	4	509.08	4	210.71	4	260.999
Random-50-70	50	860	3	5	–	3	203.18	3	156.14	3	168.483
Random-50-80	50	980	3	3	381.44	3	90.813	3	172.01	3	235.731
Random-50-90	50	1103	2	2	248.67	2	120.87	2	36.53	2	38.206
Random-100-1	100	100	46	123	–	46	8.769	46	2.61	46	0.64
Random-100-2	100	109	46	125	–	46	11.385	46	5.3	46	0.843
Random-100-3	100	181	37	109	–	37	704.88	37	23.17	37	7.421
Random-100-4	100	206	34	99	–	34	2533.2	34	66.7	34	61.702
Random-100-5	100	231	32	101	–	35	–	32	562.64	32	164.502
Random-100-6	100	321	26	90	–	28	*	26	5806.74	26	–
Random-100-7	100	317	25	87	–	26	*	25	4493.7	25	4009.377
Random-100-8	100	317		91	–	25	*	23	*	24	*
Random-100-9	100	430		81	–	23	*	21	*	22	*
Random-100-10	100	498		76	–	22	–	19	*	20	*
Random-100-20	100	981		64	–		–	12	*	12	*
Random-100-30	100	1477			–		–	11	–	11	–
Random-100-40	100	1945			–		–	9	–		–
Random-100-50	100	2483			–		–	7	–		–

using Gurobi optimization solver, usage of Gurobi optimization solver provides solution values for some instances even when CPLEX was stopped because of the memory limitations.

Given that optimization solvers were unsuccessful in providing solution values for larger graph instances than the one with 100 vertices, designing an exact method or even metaheuristic method for solving the proposed mathematical formulation is matter of the future work.

Compliance with ethical standards

Conflict of interest The author declares that there is no conflict of interest regarding the publication of this paper.

References

Ahangar HA, Henning MA, Löwenstein C, Zhao Y, Samodivkin V (2014) Signed roman domination in graphs. J Comb Optim 27(2):241–255

Alvarado JD, Dantas S, Rautenbach D (2015a) Relating 2-rainbow domination to weak roman domination. arXiv preprint arXiv:1512.01067

Alvarado JD, Dantas S, Rautenbach D (2015b) Strong Equality of Roman and Weak Roman Domination in Trees. arXiv preprint arXiv:1507.04901

Alvarez-Ruiz M, Yero IG, Mediavilla-Gradolph T, Valenzuela J (2015) A stronger vision for roman domination in graphs. arXiv preprint arXiv:1502.03933

Burger A, De Villiers A, Van Vuuren J (2013) A binary programming approach towards achieving effective graph protection. In: Pro-

ceedings of the 2013 ORSSA annual conference, ORSSA, 2013, pp 19–30
Chang GJ, Chen SH, Liu CH (2014) Edge roman domination on graphs. arXiv preprint arXiv:1405.5622
Chapelle M, Cochefert M, Couturier JF, Kratsch D, Liedloff M, Perez A (2013) Exact algorithms for weak roman domination. In: Combinatorial Algorithms, Springer, pp 81–93
Chellali M, Haynes TW, Hedetniemi ST (2014) Bounds on weak roman and 2-rainbow domination numbers. Discrete Appl Math 178:27–32
Cockayne E, Grobler P, Grundlingh W, Munganga J, Jv Vuuren (2005) Protection of a graph. Util Math 67:19–32
Currò V (2014) The roman domination problem on grid graphs. Ph.D. thesis, Università di Catania
Dreyer PA Jr (2000) Applications and variations of domination in graphs. Ph.D. thesis, Citeseer
Henning MA (2003) Defending the roman empire from multiple attacks. Discret Math 271(1):101–115
Henning MA, Hedetniemi ST (2003) Defending the roman empire a new strategy. Discret Math 266(1):239–251
Klobučar A, Puljić I (2014) Some results for roman domination number on cardinal product of paths and cycles. Kragujev J Math 38(1):83–94
Lai YL, Lin CT, Ho HM (2011) Weak roman domination on graphs. In: Proceedings of the 28th workshop on combinatorial mathematics and computation theory, Penghu University of Science and Technology, Penghu, Taiwan, May 27–28, 2011, pp 224–214
Liedloff M, Kloks T, Liu J, Peng SL (2005) Roman domination over some graph classes. In: Graph-Theoretic Concepts in Computer Science, Springer, pp 103–114
Liu CS, Peng SL, Tang CY (2010) Weak Roman Domination on Block Graphs. In: Proceedings of The 27th workshop on combinatorial mathematics and computation theory, Providence University, Taichung, Taiwan, April 30–May 1, 2010, pp 86–89
Pushpam PRL, Kamalam M (2015) Efficient weak roman domination in myscielski graphs. Int J Pure Eng Math (IJPEM) 3(2):93–100
Pushpam PRL, Kamalam M (2015) Efficient weak roman domination in graphs. Int J Pure Appl Math 101(5):701–710
Pushpam PRL, Mai TM (2011) Weak roman domination in graphs. Discuss Math Graph Theory 31(1):161–170
ReVelle CS, Rosing KE (2000) Defendens imperium romanum: a classical problem in military strategy. Am Math Mon 107(7):585–594
Shang W, Hu X (2007) The roman domination problem in unit disk graphs. In: Computational Science–ICCS 2007, Springer, pp 305–312
Song X, Yang J, Xie Y (2011) Weak Roman domination in 2xn grid graphs. J Henan Univ (Nat Sci) 41(1): 4–9 (**in Chinese**)
Song X, Bian J, Yin W (2013) Six safe grades on weak roman domination. J Henan Univ (Nat Sci) 5:002
Stewart I (1999) Defend the roman empire!. Sci Am 281:136–138
Targhi M, Rad NJ, Moradi MS (2012) Properties of independent roman domination in graphs. Australas J Combin 52:11–18