Communications in Mathematics and Applications
Vol. 15, No. 5, pp. 1541–1560, 2024
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v15i5.2840

http://www.rgnpublications.com

Research Article

Algorithm Using LPP for Domination and Fractional Domination Number of Some Graphs

Mahesh Sarada*[1], Rekha Jain[1] and Ganesh Mundhe[2]

[1] *Department of Mathematics, Medi-Caps University, Indore, Madhya Pradesh, India*
[2] *Department of Mathematics, Army Institute of Technology, Savitribai Phule Pune University, Pune, Maharashtra, India*
***Corresponding author:** mahesh.sarada@gmail.com

Received: August 2, 2024 **Accepted:** November 11, 2024

Abstract. In the given study to find the connected graph's domination number and fractional domination number we have used *Linear Programming Problem* (LPP) formulation. We have design an algorithm using LPP formulation and linear programming solver to obtain the smallest size of dominating set for cycle graph, complete graph, wheel graph, bipartite graph, tree graph, n-connected graph. The basic steps of the simplex method for solving a minimization and maximization LPP problem. Generalized algorithm to find fractional domination number of any graph. The fractional domination number of a graph has a wide range of applications in graph theory and related fields, and it is an important parameter to consider when analysing and designing networks and algorithms.

Keywords. Dominating set, Fractional domination number, LPP formulation, Algorithm

Mathematics Subject Classification (2020). 05C72, 05C85

1. Introduction

Let $G=(V,E)$ be a graph with a subset D of V is called a dominating set of graph G if each vertex in $V-D$ is connected to at least one vertex in D. The size of the smallest dominance set determines a graph's domination number. The fractional domination number, on the other hand, allows for a real-valued weight to be assigned to each vertex in the graph. A fractional dominating set is a set of vertices with non-negative weights by considering sum of the weights of all vertices and their neighbours is at least one. The fractional domination number of a graph

is the minimum sum of weights of a fractional dominating set. The fractional domination number of a graph is a variant of the domination number, which measures the smallest fraction of vertices in a graph that can be dominated so that each vertex in the dominant set is either adjacent to or dominated by other vertices. In 1980, Cockayne *et al*. [3] presented the idea and has since been extensively studied in graph theory and related fields. One of the early results in the area is a characterization of graphs with fractional domination number 1, which is equivalent to the domination number being 1. This result is due to Haynes *et al*. [10] who also proved that computing the fractional domination number is NP-hard.

In 1985, Fink and Jacobson [5] introduced generalization of the concept of domination and independence in graphs. Further, Chellali *et al*. [2] have provided a survey result on k-domination and k-independence in graphs. They also established a number of bounds and results for this variant of the problem. Lan and Chang [11] have demonstrated a linear time approach for the k-domination issue on networks where every block represents a full bipartite graph, a clique, or a cycle. All graphs include trees, block graphs, cacti and block-cactus graphs. Merouane and Mustapha [12] provided a polynomial-time approach for determining each nontrivial tree's dominator chromatic number. The observations of Fricke *et al*. [7] have offered a technique for determining the maximum n-independent set S and the total n-dominating set D in a connected graph with at least $p \geq 2n+1$ vertices where p is the order of graph.

The fractional domination number has also been studied in several graph classifications, including trees, planar graph, and hypercube. In particular, research has demonstrated that the fractional dominance number of a tree is always at most one. In recent years, researchers have also investigated fractional versions of other graph parameters, such as the independent domination number and the total domination number. There has also been interest in the computational complexity of computing these parameters, as well as in algorithms for finding them in various types of graphs. Overall, the study of fractional domination numbers and research on related parameters is also on going in the field of graph theory.

Fisher [6] has given relation between fractional dominating number of G and fractional total dominating number of graph complement. They have defined fractional domination number as 1 is the vector of all ones and 0 is the vector of all zeros. Now, $A(G)$ is adjacency matrix of graph and I be the identity matrix. The fractional dominating number $\gamma_f(G)$ is objective function value of linear programming problem:

$$\begin{aligned} &\text{objective function minimize } 1^T x, \\ &\text{subject to constraints } (A(G)+I)x \geq 1, \text{ and } x \geq 0. \end{aligned} \tag{1.1}$$

Same like this the fractional total dominating number $\Gamma_f(G)$ be the value of linear programming problem of objective function minimize $1^T x$,

$$\text{subject to constraints } (A(G)x) \geq 1, \text{ and } x \geq 0. \tag{1.2}$$

Solution to equation is fractional dominating number with non-negative weights on vertices whose sum in any closed neighbourhood is at least one. By forcing x to have integer entries transforms equation into integer program for domination number of graph. Solution to equation is fractional total dominating number with non-negative weights on vertices whose sum in any open neighbourhood is at least one. By forcing x to have integer entries transforms equation into integer program for upper domination number of graph.

The second section contains basic concepts of dominating set, domination number, fractional domination number and fractional packing number which are relevant in later sections as well as a few theorems will be covered. Third section includes an algorithm for dominating set and fractional dominating number of graph using LPP formulation. Specific graphs like cycle graph, complete graph, wheel graph, bipartite graph, tree graph and results for generalized graphs. Fourth section contains applications which are given for analysing, designing networks, algorithms and optimization problems.

2. Fundamental Definitions

The fundamental meanings of fractional domination number, fractional domination function, and dominating set are provided in [8] and [10] as follows:

2.1 Dominating Set and Domination Number

Let $G=(V,E)$ is graph with V as vertex set and E as edge set. A subset D of V is called a dominating set of graph G if each vertex in $V-D$ is connected to at least one vertex in D. For every vertex $u\in V-D$, $d(u,D)=1$ or $N[D]=V$. A domination set D is known as *minimal dominating set* (MDS) if no proper subset of D is a dominating set of G. The size of smallest dominating set of graph G is called the domination number of graph G and it is denoted by $\gamma(G)$. The maximum cardinality of *minimal dominating set* (MDS) of graph G is called upper domination number of G and it is denoted by $\gamma(G)$. The domination number noted by $\gamma(G)$ and the upper domination number noted by $\gamma(G)$ are defined as:

$$\gamma(G)=\min\{|D|:D \text{ is MDS of } G\} \quad \text{and} \quad \Gamma(G)=\max\{|D|:D \text{ is MDS of } G\}.$$

2.2 Fractional Dominating Function

A dominating function f to be any $f:V(G)\to[0,1]$ is function of G which allocates the values for each vertex $v\in V(G)$ in the unit interval $[0,1]$. The given function f is called a fractional dominating function if for every vertex $v\in V(G)$, $f(N[v])=\int_{v\in N[v]}f(v)\geq 1$. It denotes the total value of the vertices in the closed neighborhood of $v\in V(G)$ such that $N[v]$ is at least one, i.e., $(N[v])\geq 1$ (since then any vertex $v\in V(G)$ is in the closed neighborhood of at least one vertex in D, where D is subset of vertex set V). For any $v\in V(G)$, define a function $f:V(G)\to[0,1]$ by $f(v)=\frac{1}{\delta+1}$, where $\delta=\delta(G)$ minimum degree of graph then, $f(N[v])=\frac{\deg v+1}{\delta+1}\geq\frac{\delta+1}{\delta+1}=1$. So $f(N[v])\geq 1$, for all $v\in V(G)$ and f is a fractional dominating function.

2.3 Fractional Domination Number

The dominating function f is called a *minimal fractional dominating function* (MFDF) if there does not exist a dominating function $g\neq f$ for which $g(v)\leq f(v)$ for all $v\in V(G)$ equivalently f is an *minimal fractional dominating function* (MFDF) if for every vertex v with $f(v)>0$, there exist a vertex $w\in N[v]$ such that $\int_{v\in N[w]}f(v)=1$. If there is a vertex v for which given condition is not true means every vertex in the closed neighborhood of v obeys $(N[v])>1$, then we can decrease $f(v)$ to obtain a smaller fractional dominating function and so f is not a *minimal fractional dominating function* (MFDF).

The fractional domination number of G denoted by $\gamma_f(G)$ and the upper fractional domination number of G denoted by $\Gamma_f(G)$ are defined as,

$$\gamma_f(G) = \min\{|f| : f \text{ is an MFDF of } G\}, \quad \Gamma_f(G) = \max\{|f| : f \text{ is an MFDF of } G\},$$

where $|f| = \int_{v\in V} f(v)$. Fractional domination number as 1 be the vector of all ones. Let 0 be the vector of all zeros. Let $A(G)$ be the adjacency matrix. The fractional domination number $\gamma_f(G)$ is the value of linear program:

objective function minimize $1^T x$,

subject to conditions $(A(G)+I)x \geq 1$, and $x \geq 0$.

Some preliminary known results:

Theorem 2.1 ([10]). *Let for any graph G, we have* $\left\lceil \frac{n}{1+\Delta(G)} \right\rceil \leq \gamma(G) \leq n - \Delta(G)$.

Theorem 2.2 ([10]). *Let for any graph G, we have* $\frac{n}{1+\Delta(G)} \leq \gamma_f(G) \leq \frac{n}{1+\delta(G)}$, *where $\Delta(G)$ is maximum degree of graph and $\delta(G)$ is minimum degree of graph.*

Theorem 2.3 ([10]). *For any graph G, we have $\gamma_f(G) = 1$ if and only if $\Delta(G) = n-1$.*

Theorem 2.4 ([4]). *If G is an r-regular graph then $\gamma_f(G) = \frac{n}{r+1}$.*

2.4 Fractional Packing Number

Let $G = (V,E)$ be a graph and let a function $f : V(G) \to [0,1]$ be a function of G which assigns values for each vertex $v \in V(G)$ in the unit interval $[0,1]$. The function f is fractional packing function of G if for every vertex $v \in V(G)$, $f(N[v]) \leq 1$ that is, the sum of values $f(w)$ for every vertex $w \in N[v]$ is at most one. A maximal fractional packing function is fractional packing function for which every vertex $v \in V(G)$, $f(v) < 1$ that implies there exist $u \in N[v]$ such that $f(N[u]) = 1$. The lower fractional packing number of G, $p_f(G)$ and the upper fractional packing number of G, $P_f(G)$ are defined by

$$p_f(G) = \min\{|f| : f \text{ is a maximal fractional packing function of } G\}$$

and

$$P_f(G) = \max\{|f| : f \text{ is a maximal fractional packing function of } G\}.$$

3. Algorithm using LPP for Domination and Fractional Domination Number of Some Graphs

3.1 LPP Formulation of Given Graph

Let $G = (V,E)$ be an undirected graph with V is set of vertices and E is set of edges.

Fractional Domination

Primal $\gamma_f(G) = \min \sum_{i=1}^{n} x_i$

subject to $N.X \geq \overrightarrow{1}_n$, $x_i \geq 0$

Fractional Packing

Dual $P_f(G) = \max \sum_{i=1}^{n} x_i$

subject to $N.X \leq \overrightarrow{1}_n$, $x_i \geq 0$

where N is the matrix associated with a graph G.

In Figure 1, we have $\gamma_f(G) = \frac{3}{2} = P_f(G)$.

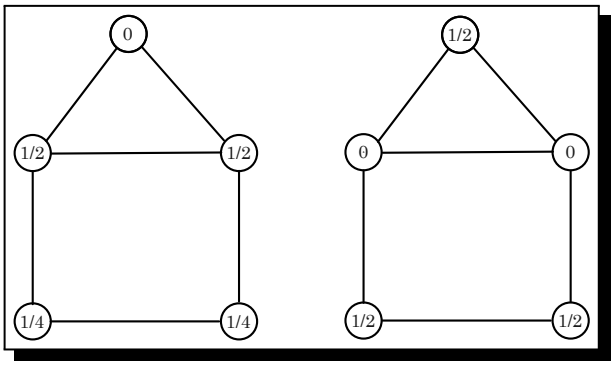

Figure 1. Linear Programing Formulation

When matrix N associated with graph G then one can also write, $\gamma_f(G) = (N, \text{MIN}, \overrightarrow{1}_n, \overrightarrow{1}_n, R^+)(G)$ and $P_f(G) = (N, \text{MAX}, \overrightarrow{1}_n, \overrightarrow{1}_n, R^+)(G)$ are duals. All of the pairs of graph theoretic LP-dual problems in this have optimal solutions. We can also write the definition as decision variable $f(v) = x_v$, binary decision variable indicating whether vertex v is dominated. $f(v) = x_v = 1$ if vertex v is dominated and $f(v) = x_v = 0$ otherwise.

Objective function: Minimize the total number of vertices dominated so

Minimize: $\sum_{v \in V} x_v$

Constraints:

(i) Every vertex v must be dominated $\int_{v \in N[v]} f(v) \geq 1$ for all $v \in V(G)$ where $N[v]$ is the closed neighborhood of vertex v means set of vertices adjacent to v.

(ii) Binary decision variables $0 \leq f(v) \leq 1$, for all $v \in V(G)$

with these definitions the LPP representation for the fractional domination number of graph is given by

Minimize: $\sum_{v \in V} x_v$

subject to (i) $\sum_{v \in N[v]} x_v \geq 1$, for all $v \in V(G)$

(ii) $0 \leq x_v \leq 1$, for all $v \in V(G)$

This LPP formulation aims to minimize the total number of vertices that are dominated subject to the constraints that every vertex must be dominated and the decision variables must be binary. Solving this LPP provides the fractional dominating number of graph.

3.2 Closed Neighborhood Domination

Assume each of the vertices in graph G of (Figure 2) represents a town, each of which supports a security service such as a fire station. Further assume that each town v can be serviced (multiple-alarm fires) by any service station in its closed neighborhood $N[v]$. Note for the graph G that $N.\overrightarrow{1_{10}} = [2,4,4,4,2,2,4,4,4,2]^t = \overrightarrow{1}_n$. If we place two service stations at each of v_2, v_4, v_7, v_9 then $N.[0,2,0,2,0,0,2,0,2,0]^t \geq \overrightarrow{1}_n$ (or it equals $\overrightarrow{1}_n$). Thus, every town is serviced as well with the 8 stations so arranged as it is with the 10 arranged as one per town. Thus, closed neighbourhood domination function $\gamma_f(G) = (N, \text{MIN}, \overrightarrow{1}_n, \overrightarrow{1}_n, R^+)(G)$ represents the minimum number of service facilities one can place on $V(G)$ so that each vertex location is serviced by

as many facilities as it is by the function $f : V(G) \to [0,1]$ with $f(v_i) = 1$ for $1 \le i \le n$, that is $f(N[v_i]) \ge |N[v_i]|$, for $1 \le i \le n$.

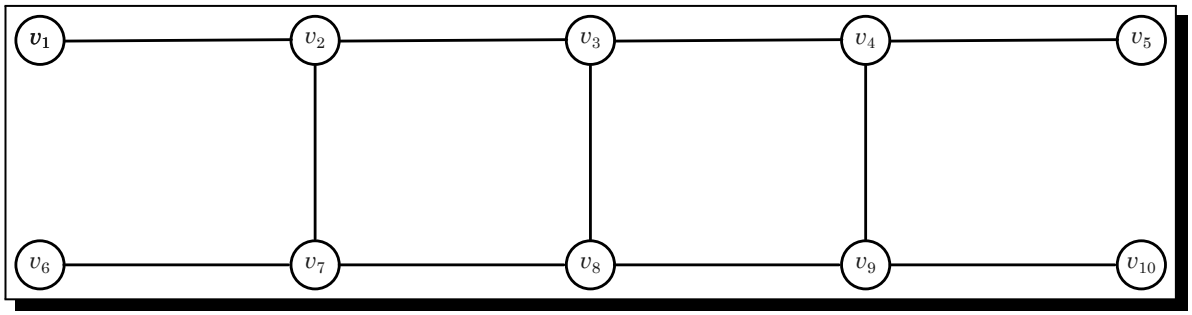

Figure 2. Closed Neighbourhood Domination

On the other hand, suppose each town supports any sickening facility such as a garbage dump or noisy airport, traffic density. For example, one assumes that the traffic density at each airport or other roads is equivalent and the noise affects that location and its immediate neighbors. Note for the graph G that we can use its dual of closed neighborhood packing function in which the objective is to place as many facilities on $V(G)$ as possible with each $N[v_i]$ containing at most $|N[v_i]|$ of them.

3.3 Algorithm for Solving LPP Problem using Simplex Method

The basic steps of the simplex method for solving a minimization and maximization LPP problem. However, keep in mind that the simplex method can become more complex with additional features such as handling equality constraints, handling unbounded solutions, and dealing with degeneracy. Following are the steps for solving a minimization and maximization LPP problem using simplex method.

Algorithm for solving a minimization and maximization LPP problem

1. Set A: Matrix of coefficients for constraints
2. Set b: Vector of constants for constraints
3. Set c: Coefficients of the objective function
4. Initialization, initialize table T with A, b, c
5. While not converged do select pivot column
6. Select the most negative coefficient in the objective function row (c_j)
7. If no negative coefficients
8. Then solution found
9. Break
10. Select pivot row
11. Compute the ratios (b_i/A_{ij}) for each row i where $A_{ij} > 0$
12. Select the smallest non-negative ratio as the pivot row
13. Update table T

14. Perform pivot operation to make the pivot element 1 and all other elements in the pivot column 0
15. Extract solution
16. Extract solution from table T
17. Return solution

The *Linear Programming Problem* (LPP) for an adjacency matrix typically entails determining the best way to solve a linear objective function given a set of linear constraints. The adjacency matrix represents a graph where each node is represented by a row and column in the matrix, and the entries in the matrix indicate the edges between nodes.

Here to formulate the LPP problem for adjacency matrix for maximizing and minimizing the solution we need to modify objective function and condition on constraints to maximize and minimize the solution.

Objective function: Maximize $Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$,

Minimize $Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n$,

where $c_1, c_2, \ldots, c_n$ are the weights assigned to each node in the graph, and $x_1, x_2, \ldots, x_n$ are binary decision variables representing whether a node is included in the solution or not.

Constraints: The constraints ensure that the solution represents a valid graph with no disconnected nodes and no cycles.

Each node can only be included once $x_1 + x_2 + \ldots + x_n = 1$ if there is an edge between two nodes, they must both be included $x_i + x_j \geq 1$ for all (i, j) pairs, where $A_{ij} = 1$. There can be no cycle in the solution $x_i + x_j \leq 1$ for all (i, j) pairs where there is cycle in the solution.

The decision variables x_i are binary, taking the value of 1 if node i is included in the solution and 0 otherwise. The objective function Z is maximized by selecting the set of nodes that have the highest weights while satisfying the constraints. The objective function Z is minimized by selecting the set of nodes that have the lowest weights while satisfying the constraints. The solution of this LPP will provide a valid graph with the maximum weight and minimum weight.

3.4 Bounds in Terms of Order

An obvious upper bound on the domination number is the number of vertices in the graph. Since at least one vertex is needed to dominate a graph, we have $1 \leq \gamma(G) \leq n$ for every graph of order n. Both of these bounds are sharp. A graph obtains the lower bound if and only if it has vertex of degree $n-1$, and it achieves the upper bound if and only if the graph $G = \overline{K}_n$, that is G is set of isolated vertices. Note that each isolated vertex must be in every dominating set. For graphs without isolated vertices, the upper bound is much improved in a classical result.

Theorem 3.1. *If a graph G has no isolated vertices, then $\gamma(G) \leq \frac{n}{2}$.*

Theorem 3.2. *For graph G with even order n and has no isolated vertices, $\gamma(G) = \frac{n}{2}$ if and only if the components of G is the cycle C_4.*

Theorem 3.3. *Consider next the class of all minimal dominating sets in a graph. The tree T in* (*Figure* 3) *has minimal dominating sets of several different cardinalities like* $\{4,5\}$, $\{4,6,7\}$, $\{1,2,3,5\}$, *and* $\{1,2,3,6,7\}$. *In particular,* $\gamma(T)=2$ *and* $\Gamma(T)=5$.

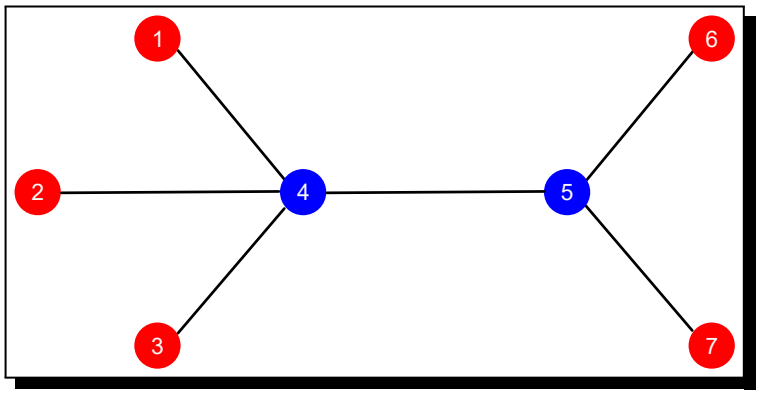

Figure 3. A tree with minimal dominating set

3.5 LPP Using a Linear Programming Solver to Obtain Upper Domination Number of Graph

We refer [13] for graph $G=(V,E)$ with the vertex set $\{V_1,V_2,V_3,V_4,V_5\}$, the adjacency relation is defined by $V_1\to V_2$, $V_1\to V_3$, $V_3\to V_4$, $V_2\to V_4$, $V_4\to V_5$, where V_5 is pendent vertex. If we take vertex set is $\{V_1,V_4\}$ which forms dominating set of the graph with the minimum size of 2. Therefore, the dominating number of given graph $\gamma(G)=2$. To find the upper dominating number of this graph $\Gamma(G)$ using linear programming, we can follow the steps:

(1) Let x_1,x_2,x_3,x_4, and x_5 be binary decision variables representing whether vertex is included or not included in the dominating set.

(2) The objective function is to Minimize $z=x_1+x_2+x_3+x_4+x_5$, which represents the smallest size of the dominating set.

(3) The conditions on constraints are

(a) If vertex x_i, where $i=1,2,3,4,5$ is not in the dominating set, then at least one of its neighbours must be in set:

- $x_1+x_3\geq 1$ (this constraint ensures that either vertex V_1 or vertex V_3 (or both) must be connected to some other vertex).
- $x_2+x_4\geq 1$ (this constraint ensures that either vertex V_2 or vertex V_4 (or both) must be connected to some other vertex).
- $x_1+x_2+x_4\geq 1$ (this constraint ensures that at least one of the vertex V_1, V_2, or V_4 must be connected to some other vertex).
- $x_3+x_4\geq 1$ (this constraint ensures that either vertex V_3 or vertex V_4 (or both) must be connected to some other vertex).
- $x_4+x_5\geq 1$ (this constraint ensures that either vertex V_4 or vertex V_5 (or both) must be connected to some other vertex).

(b) If vertex x_i where $i=1,2,3,4,5$ is in the dominating set, then it does not need to be adjacent to any other vertex in the set:

- $x_1 + x_2 \leq 1$ (this constraint means that either vertex V_1 or vertex V_2 (or neither) can have an outgoing edge, but not both. It ensures that there is at most one edge leaving vertex V_1 and vertex V_2 combined).
- $x_3 + x_4 \leq 1$ (similarly, this constraint means that either vertex V_3 or vertex V_4 (or neither) can have an outgoing edge, but not both. It ensures that there is at most one edge leaving vertex V_3 and vertex V_4 combined), and
  $x_5 \leq 1$ (this constraint ensures that vertex V_5 can have at most one outgoing edge. Since it has no other adjacent vertex in the given graph, it can have either zero or one outgoing edge).
- These constraints effectively enforce that each vertex has a maximum of one outgoing edge, ensuring acyclic behaviour in the graph. They prevent the formation of loops or cycles, making the graph undirected acyclic graph.

(c) Each variable x_i must be binary means $x_i \in \{0,1\}$.

(4) Solve the resulting LPP using a graphical method or linear programming solver to obtain the minimum size of a dominating set. We get the solution like

$$x_1 = 0,\ \ x_2 = 1,\ \ x_3 = 0,\ \ x_4 = 1,\ \ x_5 = 1.$$

Therefore, $z = 3$.

This solution corresponds to the set of vertices $\{V_2, V_4, V_5\}$, which form dominating set of the given graph with the minimum size with 3. Therefore upper domination number of graph is $\Gamma(G) = 3$.

3.6 LPP Using a Linear Programming Solver to Obtain the Smallest Size of Dominating Set of Cycle Graph

Consider the cycle graph C_n with n vertices, where $n \geq 3$, to find the smallest size of dominating set of this graph using linear programming, we can follow the steps.

Algorithm to find the smallest size of dominating set of cycle graph

1. Let $x_1, x_2, x_3, x_4, \ldots, x_n$ be binary decision variables representing whether a vertex is included in the dominating set or not. $x_i = 1$ if vertex i is in the dominating set, and $x_i = 0$ otherwise.
2. Objective function is minimize $z = x_1 + x_2 + x_3 + \cdots + x_n$, which represents the size of the dominating set. As n is the number of vertices in the cycle graph. This represents the total number of vertices in the dominating set.
3. The constraints are
   a. If vertex x_i where $i = 1, 2, 3, \ldots, n$ is not in the dominating set, then at least one of its neighbours must be in set:

      $x_{i-1} + x_{i+1} + x_i \geq 1$, for $i = 2, 3, \ldots, n-1$ (making certain that every vertex is either included in the dominating set or close to one.)
   b. If vertex x_i, where $i = 1, 2, 3, \ldots, n$ is in the dominating set, then it does not need to be adjacent to any other vertex in the set:

$x_i + x_{i-1} \leq 1$, for $i = 2,3,\ldots,n-1$ and $x_1 + x_n \leq 1$ (since the end points are adjacent to each other)

c. Each variable x_i must be binary means $x_i \in \{0,1\}$.

4. Solve the resulting LPP using a graphical method or linear programming solver to obtain the minimum size of a dominating set. We may get the solution like for cycle graph G, we have $\left\lceil \frac{n}{1+\Delta(G)} \right\rceil \leq \gamma(G) \leq n - \Delta(G)$ or $Z = \lceil \frac{n}{3} \rceil$.

We can follow the step:

1. Enter number of vertices in the cycle graph.
2. Set n as integer domination_number
3. If (n % 3 == 0) then domination_number = (n/3)
4. Else domination_number = ceil(n/3.0)
5. The dominating number of cycle graph with n vertices domination_number endl
6. Return 0

Output: Enter number of vertices in the given cycle graph: 7

The dominating number of the cycle graph with 7 vertices is 3.

The user is prompted to enter the number of vertices in the cycle graph. If the number of vertices is even, then the domination number is (n/3). This is because every vertex is adjacent to other two vertices to form a dominating set. If the number of vertices is odd, then domination number is $\lceil \frac{n}{3} \rceil$.

3.7 Minimum Size of a Dominating Set of Complete Graph

Consider the complete graph K_n with n vertices, where $n \geq 1$.

To find smallest size of dominating set of this graph using linear programming, we can follow the steps.

Algorithm to find the smallest size of dominating set of complete graph

---

1. Let $x_1, x_2, x_3, x_4, \ldots, x_n$ be binary decision variables representing whether a vertex is included in the dominating set or not.
2. The objective function is to Minimize $z = x_1 + x_2 + x_3 + \cdots + x_n$, which represents the size of dominating set. As n is the number of vertices in the complete graph. This represents the total number of vertices in the dominating set.

   a. If vertex x_i where $i = 1,2,3,\ldots,n$ is not in dominating set, then at least one of its neighbours must be there in set:

   $x_1 + x_2 + \cdots + x_{i-1} + x_{i+1} + \cdots + x_n \geq 1$, for $i = 1,2,3,\ldots,n$.

   b. If vertex x_i, where $i = 1,2,3,\ldots,n$ is in the dominating set, then it does not need to be adjacent to any other vertex in the set:

   $x_i + x_j \leq 1$ for all $i \neq j$

   c. Each variable x_i must be binary means $x_i \in \{0,1\}$.

3. Solve the resulting LPP using a graphical method or linear programming solver to obtain the minimum size of a dominating set. We may get the solution like $z = 1$.

Every vertex in a complete graph has $(n-1)$ neighbors, so any vertex chosen for the dominating set will dominate all other vertices. Therefore, the minimum size of a dominating set is 1, and any vertex can be chosen as the dominating vertex.

1. Enter the number of vertices in the complete graph as int n
2. Enter the degree of each vertex as int r
3. Set domination_number $= (n/(r+1))$
4. The dominating number of complete graph with n vertices domination_number end
5. Return 0.

Output: Enter number of vertices in complete graph: 5

Enter degree of each vertex: 4

The domination number of the complete graph with 5 vertices is 1.

The user is prompted to enter the number of vertices in the complete graph. The domination number of a complete graph with n vertices is given by the formula $(n/(r+1))$. The program outputs the domination number of the complete graph.

3.8 Minimum Size of Dominating Set of Wheel Graph

Consider the wheel graph W_n with $n+1$ vertices, where $n \geq 2$. The wheel graph consists of a cycle graph C_n with an additional vertex as hub that is adjacent to every vertex of the cycle.

To find the minimum size of a dominating set of this graph using linear programming, we can follow the steps.

Algorithm to find the smallest size of dominating set of wheel graph

---

1. Let $x_1, x_2, x_3, x_4, \ldots, x_{n+1}$ be binary decision variables representing whether a vertex is included in the dominating set or not.
2. The objective function is to Minimize $z = x_1 + x_2 + x_3 + \cdots + x_{n+1}$, which represents the size of dominating set. As n is number of vertices in the wheel graph. This represents the total number of vertices in the dominating set.

   The constraints are:

   a. $x_{i-1} + x_{i+1} + \cdots + x_{n+1} \geq 1$, for $i = 2, 3, \ldots, n$ where we use the cyclic notation that $x_0 = x_n$ and $x_{n+1} = x_1$.
   b. $x_1 + x_{n+1} \geq 1$ since the hub is adjacent to every vertex of the cycle
   c. If vertex i is in the dominating set, then it does not need to be adjacent to any other vertex in the set
   d. $x_{i-1} + x_{i+1} + x_i \leq 1$, for $i = 2, 3, \ldots, n$ where we use the cyclic notation that $x_0 = x_n$ and $x_{n+1} = x_1$
   e. $x_1 + x_2 + x_{n+1} \leq 1$, since the hub is adjacent to every vertex of the cycle
3. Each variable x_i must be binary means $x_i \in \{0, 1\}$

4. Using LPP graphical method or programming solver we obtain the optimal solution like

$$x_i = 1, \text{ for } i = 1,2,\ldots,n+1, \quad x_i = 0, \text{ for } i = 2,3,\ldots,n, \; Z = 1.$$

This solution corresponds to the set of vertices $\{1, n+1\}$ which forms a dominating set of the wheel graph with minimum size. This set includes the hub and one vertex from the cycle.

1. Enter the number of vertices in the wheel graph as int n;
2. Set int domination_number;
3. if ($n >= 4$) then domination_number = 1;
4. else domination_number = 0;
5. The domination number of the wheel graph with n vertices is domination_number endl;
6. return.

Output: 1) Enter the number of vertices in the wheel graph: 6
The domination number of the wheel graph with 6 vertices is 1.
2) Enter the number of vertices in the wheel graph: 3.
The domination number of the wheel graph with 3 vertices is 0.

The user is prompted to enter number of vertices in the wheel graph. If there will be only 3 vertices, then the domination number is 0. This is because for wheel graph we need at least 4 vertices. If there are more than 3 vertices, then the domination number is 1. This is because there is only one dominating vertex in the central cycle of the wheel, and each spoke can be dominated by central vertex in the cycle.

3.9 Minimum Size of a Dominating Set of Bipartite Graph

Consider the bipartite graph (Figure 4) with vertex set $V = U \cup W$, where U and W are two disjoint sets of vertices, and each edge of the graph connects a vertex in U to a vertex in W.

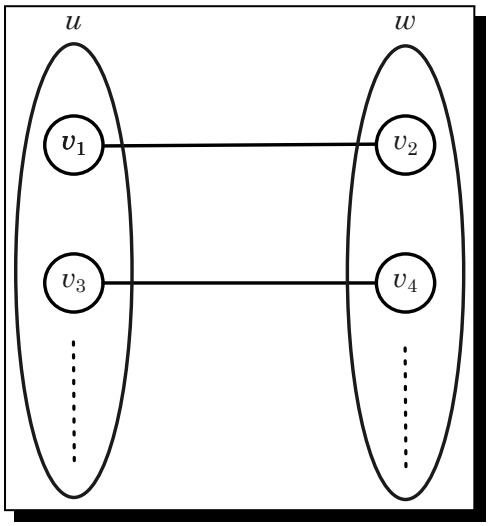

Figure 4. Bipartite graph

To find minimum size of dominating set of this graph using linear programming, we can follow the steps.

Algorithm to find the smallest size of dominating set of bipartite graph

1. Let x_u and x_w be binary decision variables representing whether a vertex in U or W, respectively, is included in the dominating set or not.
2. The objective function is to Minimize: $Z = xu_1 + xu_2 + \ldots + xu_n + xw_1 + xw_2 + \ldots + xw_m$, which represents the size of the dominating set.
3. The constraints are:
   a. If vertex i in U is not in the dominating set, then at least one of its neighbors in W must be in the set, $x_i \leq xw_j$ for all edges (i,j) in G
   b. If vertex j in W is not in the dominating set, then at least one of its neighbors in U must be in the set, $x_j \leq xu_i$ for all edges (i,j) in G
   c. Each variable x_i must be binary, $x_i \in \{0,1\}$, for $i = 1,2,\ldots,n+m$.
4. Solve the resulting LPP using a linear programming solver to obtain the minimum size of a dominating set. $x_i = 1$, for all i in U such that there is at least one edge incident to i, $x_j = 0$, for all j in $Z = |U|$.

This solution corresponds to the set of vertices in U that are incident to at least one edge, which form a dominating set of the bipartite graph G with the minimum size. Therefore, the minimum size of a dominating set of the bipartite graph G is $|U|$, which is achieved by choosing all vertices in U that are incident to at least one edge.

3.10 Minimum Size of a Dominating Set for Tree Graph

Let's consider a tree graph T with 9 vertices shown in (Figure 5).

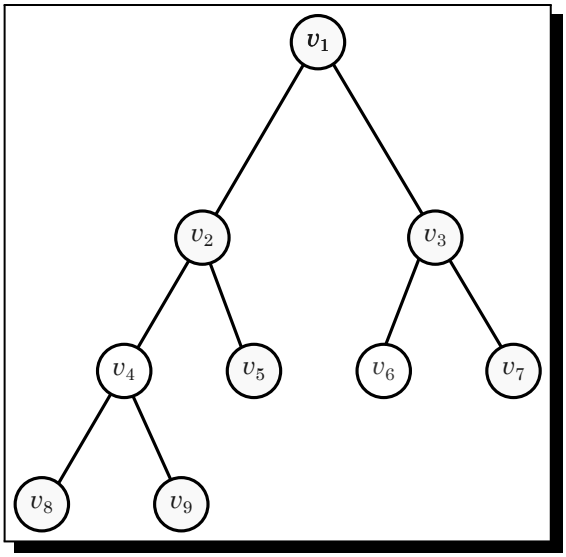

Figure 5. Tree graph

To find minimum size of dominating set of given graph using linear programming, we can follow the steps.

Algorithm to find the smallest size of dominating set of tree graph

1. Let x_i be binary decision variables representing whether a vertex i is included in the dominating set or not.
2. The objective function is to Minimize $z = x_1 + x_2 + x_3 + \cdots + x_n$, which represents the size of the dominating set.
3. The constraints are:
   a. If vertex i is not in the dominating set, then at least one of its neighbors must be in the set: $x_i + x_j \geq 1$, for all (i, j) in T
   b. Each variable x_i must be binary, $x_i \in \{0, 1\}$, for $i = 1, 2, \ldots, n$
4. Solve the resulting LPP using a linear programming solver to obtain minimum size of dominating set. Using LPP linear programming solver, we can obtain the optimal solution.

$x_i = 1$, for all leaves i in T $x_i = 0$, for all non-leaves i in T so $Z = |\text{leaves}(T)|$. This solution corresponds to the set of leaves in the tree T, which form a dominating set of the tree with the minimum size. Therefore, the minimum size of a dominating set of the tree T is $|\text{leaves}(T)|$, which is achieved by choosing all leaves in the tree.

Theorem 3.4. *For any tree T, $\gamma_f(T) = \gamma(T)$.*

3.11 To Obtain Fractional Dominating Number of Regular Graph

Consider a regular graph G with n vertices and degree k (each vertex has degree k). To find the fractional domination number of this graph using linear programming, we can follow the steps.

Algorithm to find fractional dominating number of regular graph

1. Let x_i be non-negative decision variables representing the fraction of vertices in the dominating set that include vertex i.
2. The objective function is to Minimize $Z = x_1 + x_2 + \ldots + x_n$, which represents the size of the dominating set.
3. The constraints are:
   a. For every vertex i, the sum of the fractions of its neighbours that are in the dominating set must be at least 1 $\sum_{j \in N(i)} x_j \geq 1$, for all i in G
   b. Each variable $x_i = \frac{1}{k+1} = y_i$ is feasible to LPP and yields the same objective function value of $\frac{n}{k+1}$, where Each variable x_i must be non-negative $x_i \geq 0$, for all $i = 1, 2, \ldots, n$.
4. Solve the resulting LPP using a linear programming solver to obtain the fractional domination number of the graph G.Using a linear programming solver, we obtain the optimal solution: $x_i = n/k + 1$, for all i in G and $Z = 1$, if graph G has n vertices and $(n-1)$-regular graph.

This solution corresponds to the set of vertices in G, where each vertex i is assigned a fraction of $(1/k + 1)$. Since G is a k-regular graph, every vertex has the same degree k, so each vertex

is assigned the same fraction $(1/k+1)$ in the dominating set. The sum of the fractions of the vertices in the dominating set is equal to the size of the dominating set, which is the objective function that we are minimizing. Therefore, the fractional domination number of the regular graph G is 1, which is achieved by choosing each vertex with a fraction of $(1/k+1)$ in the dominating set and graph G has n vertices with $(n-1)$-regular graph.

Theorem 3.5. *For any r-regular graph G of order n, $\gamma_f(G) = \frac{n}{r+1}$.*

3.12 The Generalized Formula for Fractional Domination Number of n-Connected Graph

The fractional domination number of a n-connected graph can be defined as the minimum value of the sum of weights of a fractional dominating set, where a fractional dominating set is a set of vertices in G such that every vertex in G is either in the set or adjacent to a vertex in the set.

The generalized formula for the fractional domination number of a n-connected graph G is:

$$\gamma_f(G) = \min\left\{\sum w(v) : \{v \in V(G)\} \text{ is a fractional dominating set}\right\},$$

where $V(G)$ is the set of vertices of graph G, $w(v)$ is the weight assigned to vertex v, and the minimum is taken over all possible fractional dominating sets of G. Note that the weight function $w(v)$ must satisfy the following conditions:

1. $w(v) \geq 0$, for all $v \in V(G)$.
2. $\sum w(v) = 1$, for all fractional dominating sets.
3. If v is a cut vertex (a vertex whose removal disconnects the graph), then $w(v) = 1$.

The third condition is necessary to ensure that every cut vertex is included in the fractional dominating set, as removing a cut vertex can potentially disconnect the graph, and the remaining vertices need to be dominated by vertices in the set. Note that for $n > 2$ the graph G is necessarily 2-connected, as a graph needs to be at least 2-connected to be n-connected. Thus, the formula for the fractional domination number of a 2-connected graph can be used for any n-connected graph.

Algorithm for fractional domination number of n-connected graph (G, n)

Let $G = (V, E)$ is graph for which we want to find the fractional domination number of n-connectivity parameter.

1. Set up the linear programming problem with the addition of constraints. This constraints ensure that for every subset S of vertices with size at most n, there exist at least one vertex in S that dominates all other vertices in S. This reflects the n-connectivity property of the graph.
2. Define variables x_v for each vertex v in G, representing whether vertex v is dominated
3. Define objective function: Minimize sum of x_v, for all vertices $v \in V(G)$

   Subject to the constraints:

   a. x_v + sum of x_u for all u adjacent to $v \geq 1$ for all vertices $v \in V(G)$
   b. Sum of x_v for all vertices v in any subset S of size $\leq n$ is at least one.
   c. $0 \leq x_v \leq 1$ for all vertices $v \in V(G)$

4. Solve the linear programming problem using a linear programming solver to find the optimal solution to the problem and the objective function value obtained which represents the fractional domination number of n-connected graph.
5. Extract the fractional domination number
6. Fractional domination number = sum of objective function value obtained from the solver
7. Return fractional domination number

3.13 To Obtain Fractional Dominating Number of Any Graph

To find fractional dominating number of a graph using linear programming, we can follow the steps below:

Algorithm to obtain fractional domination number of any graph

Input: Graph $G = (V, E)$

Define $n = |V|$, the number of vertices in G. Set up linear programming problem with n variables $x(v)$, one for each vertex $v \in V$.

1. Let x_i be non-negative decision variables representing the fraction of vertices selected in the dominating set that include vertex i.
2. The objective function is to Minimize $z = x_1 + x_2 + x_3 + \cdots + x_n$ which represents the size of the dominating set.
3. The constraints are:
   a. For every vertex i, the sum of the fractions of its neighbours that are in the dominating set must be at least 1: $\sum_{j \in N[i]} x_j \geq 1$, for all i in G
   b. Each variable x_i must be non-negative $x_i \geq 0$, for all $i = 1, 2, \ldots, n$.
   c. For every vertex i we assign a fraction weight from $[0, 1]$ to satisfy the condition $\sum_{j \in N[i]} x_j \geq 1$.
4. For some graph G with n vertices, $\Delta(G)$ is maximum degree and $\delta(G)$ is minimum degree, we may use inequality $\frac{n}{1+\Delta(G)} \leq \gamma_f(G) \leq \frac{n}{1+\delta(G)}$ for finding fractional domination number.
5. Solve the resulting LPP using a linear programming solver to obtain the fractional domination number of the graph G.
6. Using a linear programming solver, we obtain the optimal solution: $x_i = \frac{1}{\deg(i)}$, for all i in G, $Z = \sum i = 1$ to $n\left(\frac{1}{\deg(i)}\right)$. Depending on $\gamma_f(G) = \min \sum_{v \in V} x(v)$, $\{x(v) \geq 0\}$ subject to $\sum_{v \in N[u]} x(v) \geq 1$ for all $u \in V$, where V is the vertex set of the graph G, $N[u]$ is the set of neighbours of vertex u, and $x(v)$ represents a non-negative real number associated with vertex v.
7. This solution corresponds to the set of vertices in G, where each vertex i is assigned a fraction of $\frac{1}{\deg(i)}$, where $\deg(i)$ is the degree of vertex i. The sum of the fractions of the vertices in the dominating set is equal to the size of the dominating set, which is the objective function that we are minimizing. Therefore, the fractional domination number of the graph G is $\sum i = 1$ to $n\left(\frac{1}{\deg(i)}\right)$, which is achieved by choosing each vertex i with

a fraction of $\frac{1}{\deg(i)}$ in the dominating set. This algorithm provides a structured way to compute the fractional domination number of any given graph using linear programming techniques.

3.14 Example of a Simple Graph Represented in GraphQL (Graph Query Language) the Schema for the Social Network Graph

The Graph Query Language (GraphQL) plays a crucial role in various applications within graph theory. Here is one key application. We have a graph representing a social network where users are nodes and friendships between users are edges.

Algorithm for the schema for the social network graph

```
# Define the schema for the social network graph
type User {
id: ID!
name: String!
age: Int
friends: [User]
}
type Query {
getUser(id: ID!): User
getAllUsers: [User]
}
type Mutation {
addUser(name: String!, age: Int): User
addFriend(userId: ID!, friendId: ID!): User
}
# Sample data representing the graph
type SampleData {
users: [User]
}
# Resolver functions to handle queries and mutations
schema {
query: Query
mutation: Mutation
}
```

In this schema

- Each User has an id, name, age, and a list of friends.
- The Query type defines operations to retrieve users and their information.
- The Mutation type defines operations to add users and create friendships between users.
- The SampleData type is used to represent sample data for demonstration purposes.
- Resolver functions are not provided in this GraphQL schema.

In a real application, resolver functions would be implemented to handle queries and mutations by fetching or modifying data from a database

1. getUser(id: ID!): Query to retrieve a user by their ID.
2. getAllUsers: Query to retrieve all users in the social network.
3. addUser(name: String!, age: Int): Mutation to add a new user to the social network.
4. addFriend(userId: ID!, friendId: ID!): Mutation to create a friendship between two users specified by their IDs.

This GraphQL schema represents a basic graph structure for a social network, where users can be queried, added, and connected through friendships.

4. Applications

The fractional domination number of a graph is a variant of the domination number that allows for the possibility of multiple dominators per vertex. Some of the applications of the fractional domination number of a graph are:

1. *Resource Allocation*: The fractional domination number of a network can be used to allocate resources among the nodes. For example, in a wireless sensor network, the fractional domination number can be used to allocate power levels to the sensors such that the network is fully covered.
2. *Network Design*: The fractional domination number of a graph can be used in network design problems, where the goal is to construct a network that covers all the nodes of the graph. This can be useful in designing wireless communication networks, sensor networks, and transportation networks.
3. *Social Networks*: The fractional domination number of a social network can be used to identify the most influential individuals in the network. This can be useful in marketing and outreach programs, as well as in identifying opinion leaders and trendsetters.
4. *Algorithm Design*: The fractional domination number of a graph can be used in the design of algorithms for various graph problems, such as the vertex cover problem, the independent set problem, and the maximum clique problem.
5. *Epidemic Control:* The fractional domination number of a contact network can be used to identify the minimum number of individuals that need to be vaccinated or isolated to prevent the spread of an epidemic.
6. *Graphs* are also used for query optimization in database languages in some specialized compilers.
7. *Game Theory:* The fractional domination number of a game graph can be used to determine the minimum number of players required to control the game. This can be useful in the analysis of cooperative and non-cooperative games.
8. *Communication Networks*: The fractional domination number of a communication network can be used to determine the minimum amount of bandwidth required to connect all the nodes in the network.

9. Graphs are extensively used to build the transportation system especially the road network. A popular example is Google maps that extensively uses graphs to indicate directions all over the world.

Overall, the fractional domination number of a graph has a wide range of applications in graph theory and related fields, and it is an important parameter to consider when analysing and designing networks and algorithms.

5. Conclusion

The study of fractional domination number has important applications in fields such as computer science, social network analysis, and communication networks, among others. The fractional domination number provides a quantitative measure of the burden of a network, and can be used to design more robust and efficient networks. In addition to its many applications outside of computer science, a graph is a widely used. We have provided an algorithm using LPP formulation and linear programming solver to obtain the smallest size of dominating set for cycle graph, complete graph, wheel graph, bipartite graph, tree graph, n-connected graph. The basic steps of the simplex method for solving a minimization and maximization LPP problem. Generalized algorithm to find fractional domination number of graphs.

Conflict of interest

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

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

References

[1] V. A. Alfred, E. H. John and D. U. Jeffrey, *The Design and Analysis of Computer Algorithms*, Addison-Wesley Publishing Company, x + 470 pages (1974).

[2] M. Chellali, O. Favaron, A. Hansberg and L. Volkmann, k-Domination and k-independence in graphs: A survey, *Graphs and Combinatorics* **28** (2012), 1 – 55, DOI: 10.1007/s00373-011-1040-3.

[3] E. J. Cockayne, R. M. Dawes and S. T. Hedetniemi, Total domination in graphs, *Networks* **10**(3) (1980), 211 – 219, DOI: 10.1002/net.3230100304.

[4] G. S. Domke, S. T. Hedetniemi and R. C. Laskar, Fractional packing, coverings and irredundance in graphs, *Congressus Numerantium* **66** (1988), 227 – 238.

[5] J. F. Fink and M. S. Jacobson, n-Domination in graphs, in: *Graph Theory with Application to Algorithms and Computer Science*, John Wiley and Sons, New York, pp. 283–300 (1985).

[6] D. C. Fisher, Fractional dominations and fractional total dominations of graph complements, *Discrete Applied Mathematics* **122** (1-3) (2002), 283 – 291, DOI: 10.1016/s0166-218x(01)00305-5.

[7] G. H. Fricke, M. A. Henning, O. R. Oellermann and C. S. Henda, An algorithm to find two distance domination parameters in a graph, *Discrete Applied Mathematics* **68**(1-2) (1996), 85 – 91, DOI: 10.1016/0166-218x(95)00057-x.

[8] F. Harary, *Graph Theory*, Addison-Wesley Publishing Company Inc., ix + 274 pages (1969).

[9] T. W. Haynes, S. T. Hedetniemi and M. A. Henning, *Topics in Domination in Graphs*, Springer Cham., viii + 545 pages (2020), DOI: 10.1007/978-3-030-51117-3.

[10] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, *Fundamentals of Domination in Graphs*, 1st edition, CRC Press, Boca Raton, 464 pages (1998), DOI: 10.1201/9781482246582.

[11] J. K. Lan and G. J. Chang, Algorithmic aspects of the k-domination problem in graphs, *Discrete Applied Mathematics* **161**(10-11) (2013), 1513 – 1520, DOI: 10.1016/j.dam.2013.01.015.

[12] H. B. Merouane and C. Mustapha, An algorithm for the dominator chromatic number of a tree, *Journal of Combinatory Optimization* **30** (2015), 27 – 33, DOI: 10.1007/s10878-013-9631-y.

[13] M. Sarada, R. Jain and G. Mundhe, Applications of the domination and fractional domination in computational biology using LPP formulation, *African Journal of Biological Sciences* **6**(5) (2024), 4340 – 4358.

[14] M. Sarada, R. Jain and G. Mundhe, Relation between the fractional domination number of some graphs and their line graphs, *Communications on Applied Nonlinear Analysis* **31**(6s) (2024), 670 – 691, DOI: 10.52783/cana.v31.1260.