Int. J. Anal. Appl. (2026), 24:167

International Journal of Analysis and Applications

Smart Monitoring of Urban Feeder Systems via Power Dominator Coloring: A Graph-Theoretic Approach for Chennai Metro

G. Navamani[1], L. Jacquline[1,*], Daniel Breaz[2,*], Kadhavoor R. Karthikeyan[3]

[1]*Department of Mathematics, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai 602105, Tamil Nadu, India*

[2]*Department of Mathematics, "1 Decembrie 1918" University of Alba Iulia, Alba Iulia, 510009, Romania*

[3]*Department of Applied Mathematics and Science, National University of Science & Technology, Muscat P.O. Box 620, Oman*

**Corresponding author:* dbreaz@uab.ro; jackinfanci@gmail.com

Abstract. Power dominator coloring is a variant of dominator coloring in which every vertex in a graph must power dominate at least one color class in a proper vertex coloring. In this study, we investigate the power dominator chromatic number, denoted by $\chi_{pd}(G)$, specifically for caterpillar graphs. We establish tight bounds for $\chi_{pd}(G)$, based on the support vertices along the spine of the caterpillar. Capitalizing on the structural properties of these graphs, such as spine and support vertex distribution, we introduce an efficient algorithm to calculate $\chi_{pd}(G)$. A linear time implementation of the algorithm using MATLAB is constructed, and its application is illustrated in a case analysis of the Chennai Metro feeder network. The results illustrate effective monitoring and optimal resource allocation, highlighting the practical utility of power dominator coloring in real-world urban transport networks.

1. Introduction

Graph domination is a significant and widely studied topic in graph theory, first characterized by Haynes [1]. A dominating set [2] of a graph G is a subgraph S of its vertex set $V(G)$ such that each vertex not in S has at least one of S's neighbors. The smallest such set is referred to as the domination number, which is represented by $\gamma(G)$. A subset S is a γ-set of G when it has attained this minimum cardinality [3]. Following this concept, Haynes et al. [4] initiated power domination, a notion of immense applications of electric power systems. Here, a graph G symbolises the network, whose nodes are the electrical nodes edges as lines of transmission.

Received: Aug. 23, 2025.

2020 *Mathematics Subject Classification.* 05C15, 05C69.

Key words and phrases. power dominator coloring; caterpillar graphs; support vertices; graph coloring algorithm; effective monitoring.

https://doi.org/10.28924/2291-8639-24-2026-167

ISSN: 2291-8639

The overall objective is to identify the minimum number of Phasor Measurement Units (PMUs) required for complete system monitoring.

For a connected graph G and a subset Y of its vertices, the monitored set $M(Y)$ is determined through the following steps:

- **Initialization:** Include all vertices in Y and their adjacent vertices in $M(Y)$.
- **Iteration:** If there exists a vertex y in $M(Y)$ such that all but one of its neighbors, say v, are already in $M(Y)$, then add v to $M(Y)$. Once this process concludes, $M(Y)$ represents the set of vertices monitored by Y.

A *power dominating set* Y of G is one for which $M(Y) = V(G)$. The smallest cardinality of such a set is denoted by $\gamma_p(G)$. Research on power domination has been extensive [4–8].

A vertex k power dominates the vertices in $M(k)$ if the following criteria are met:

(1) Every vertex in $N(k) \cup \{k\}$ is included in $M(k)$.
(2) If a vertex $c \in V(G)$ is not yet in $M(k)$, it is added if there exists a neighbor $b \in M(k)$ such that all neighbors of b, except c, are already in $M(k)$.
(3) This process is repeated iteratively until all vertices are included.

Another essential concept in graph theory is *vertex coloring*, where vertices are assigned labels (colors) such that no two adjacent vertices share the same color. A *dominator coloring* of G is a coloring in which each vertex dominates at least one color class, and the dominator chromatic number is represented as $\chi_d(G)$ [9,10]. In *power dominator coloring (PDC)*, each vertex must power dominate at least one color class within a proper coloring of G. The corresponding chromatic number is denoted by $\chi_{pd}(G)$ [11,12].

2. Preliminaries

Let $G = (V, E)$ be a graph with vertex set V and edge set E. We denote the neighborhood of a vertex v as $N(v) = \{u \in V \mid (u, v) \in E\}$. The degree of a vertex v, denoted as $d(v)$, is the number of vertices in its neighborhood, i.e., $d(v) = |N(v)|$ [13]. Caterpillar graphs, a subclass of trees, are known for their relatively simple structure, making them amenable to algorithmic solutions. A *caterpillar* is a tree in which all vertices are within distance 1 of a central path [14]. The vertices in the central path are called *spine vertices*, and the vertices adjacent to the spine vertices but not on the spine

3. Theorem and Proof

Lemma 3.1. *[12] For a path P_n, $n \geq 2$, the power dominator chromatic number is $\chi_{pd}(P_n) = 2$.*

Theorem 3.1. *Let T_c be a caterpillar graph with t support vertices in the spine. Then:*

$$\chi_{pd}(T_c) = \begin{cases} 2, & \text{if } T_c \cong P_n \\ t+2, & \text{if } T_c \text{ has at least two adjacent non-support vertices on the spine} \\ t+1, & \text{otherwise} \end{cases}$$

Proof. Let T_c be a caterpillar of order n with t support vertices on the spine. To find the power dominator coloring for T_c, consider the following cases:

Case (i): $T_c \cong P_n$.
By Lemma 3.1, $\chi_{pd}(T_c) = 2$.

Case (ii): T_c has at least two adjacent non-support vertices on the spine.
Each leaf attached to the support vertices power dominates its closed neighborhood. Assign t distinct colors to the support vertices on the spine, say $1, 2, \ldots, t$, such that each leaf attached to them is covered. The non-support vertices in the spine also power dominate at least one color class assigned to their neighboring support vertices.

Now, to color the adjacent non-support spine vertices, we need at most two additional colors, say $t+1$ and $t+2$. Assign color $t+1$ to all leaf vertices. By the definition of power dominator coloring, each vertex in T_c power dominates at least one color class. Therefore, $\chi_{pd}(T_c) = t+2$.

Case (iii): T_c has no two adjacent non-support vertices on the spine.
Assign the colors $1, 2, \ldots, t$ to the support vertices. Assign the color $t+1$ to all non-support spine vertices as well as to all leaf vertices. This ensures that each vertex in T_c power dominates at least one color class. Therefore, $\chi_{pd}(T_c) = t+1$. □

Theorem 3.3. Let T_c be a caterpillar and t be the number of support vertices in the spine of T_c. Then

$$2 \leq \chi_{pd}(T_c) \leq t+2,$$

Proof. **Lower Bound:** $\chi_{pd}(T_c) \geq 2$.
The power dominator chromatic number cannot be 1 for any caterpillar graph. For example, consider the smallest caterpillar P_2 (a path on two vertices). It is impossible to assign only one color while satisfying the power dominator coloring conditions. Thus, a single color is insufficient even in the simplest case, establishing the lower bound $\chi_{pd}(T_c) \geq 2$.

Upper Bound: $\chi_{pd}(T_c) \leq t+2$.
Assign each support vertex a unique color from 1 to t. Each support vertex then power dominates its own color class and the leaf adjacent to it. For the non-support vertices in the spine If no two of them are adjacent, they can all be colored with a single additional color $t+1$. If there exist two adjacent non-support vertices, use two more colors $t+1$ and $t+2$ to maintain proper coloring. Additionally, all leaf vertices can be assigned the color $t+1$, as they are neighbors to support vertices with distinct colors.

Therefore, in the worst case (when adjacent non-support spine vertices exist), we use $t+2$ colors. Using more than $t+2$ colors would violate the chromatic minimality condition. Hence, $\chi_{pd}(T_c) \leq t+2$. □

4. Algorithm for Power Dominator Coloring of Caterpillars

Input: A caterpillar graph $G = (V, E)$ with n vertices.
Output: Chromatic number $\chi_{pd}(G)$.

(1) Identify support vertices v_i on the spine such that $d(v_i) > 2$. Let there be k such vertices.
(2) Assign color classes $C^i(A) = \{v_i\}$ for $1 \leq i \leq k$.
(3) Let $S = V \setminus \{v_i \mid v_i \in C^i(A)\}$.
(4) Initialize two independent sets: $C^{k+1} = [\,]$, $C^{k+2} = [\,]$.
(5) Select a random vertex $u \in S$ and assign u to C^{k+1}.
(6) For each $w \in S$:
   - If $w \notin C^i(A)$ and $w \notin N(u)$ where $u \in C^{k+1}$, then append w to C^{k+1}.
   - Else, append w to C^{k+2}.
(7) Check for a path $P_2 = (x, y)$ entirely within S (i.e., without support vertices).
   - If such a path exists, then $\chi_{pd}(G) = k + 2$.
   - Else, $\chi_{pd}(G) = k + 1$.

5. Example Illustration

Consider a caterpillar graph where the spine consists of vertices $v_1, v_2, v_4, v_5, v_8, v_{10}, v_{11}, v_{12}, v_{13}$ and v_{15}, with v_2, v_5, v_8, v_{10} and v_{13} being the support vertices on the spine.

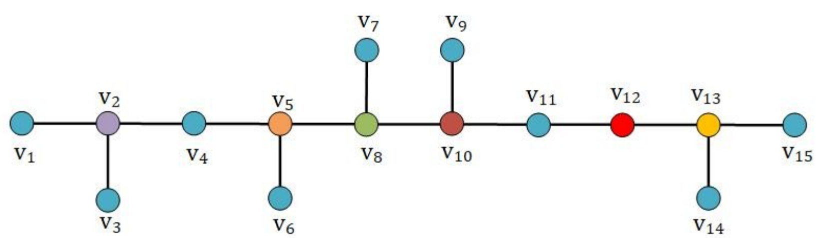

Figure 1. Caterpillar graph with two non-support vertices on the spine

Step 1: v_2, v_5, v_8, v_{10} and v_{13} are assigned distinct colors C^1, C^2, C^3, C^4, and C^5.
Step 2: $S = V \setminus \{v_2, v_5, v_8, v_{10}, v_{13}\}$.
Steps 3–4: The remaining vertices in S are distributed into C^6 and C^7 based on adjacency.
Step 5: If no path P_2 exists entirely within S, then $\chi_{pd}(G) = 6$. Otherwise, $\chi_{pd}(G) = 7$.

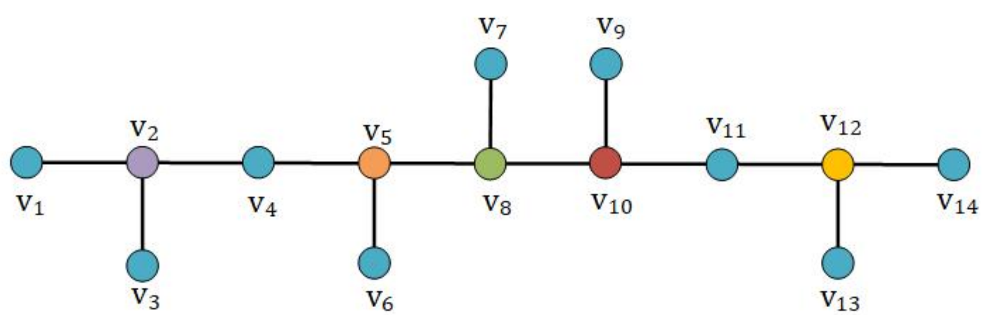

Figure 2. Caterpillar graph with no two non-support adjacent vertices in the spine

Therefore, $\chi_{pd}(G) = 7$ for Figure.1, and $\chi_{pd}(G) = 6$ for Figure.2.

6. Complexity Analysis

Time Complexity:
Finding vertices of degree greater than 2 takes $O(n)$ time. Partitioning the set S using adjacency checks also requires $O(n)$ time. Path detection in S is $O(n)$ for small caterpillar graphs, but may go up to $O(n^2)$ in general due to implementation overhead. Thus, the overall worst-case time complexity is $O(n^2)$, although it typically performs in $O(n)$ time for practical caterpillar graphs.

Space Complexity:
Storing the graph and color classes requires $O(n)$ space.

7. MATLAB Implementation of Power Dominator Coloring on Caterpillar Graphs

The Power Dominator Coloring of caterpillar graphs is implemented using the following MATLAB code. The algorithm begins by identifying support vertices on the spine and assigns a unique color class to each of them. These vertices often represent key monitoring centers, such as major metro stations. Remaining vertices are partitioned into two independent sets using a greedy coloring strategy. The process ensures adjacency constraints are respected to determine whether one or two additional colors are necessary. This technique is highly applicable to real-world infrastructure networks such as metro rail feeders and provides an efficient method for computing the power dominator chromatic number χ_{pd}.

The algorithm performs in linear or near-linear time for typical caterpillar graphs, making it suitable for large-scale applications.

```
% Example adjacency matrix for a caterpillar graph
adjMatrix = [
    0 1 0 0 0 0 0;
    1 0 1 1 0 0 0;
    0 1 0 0 0 0 0;
    0 1 0 0 1 0 0;
    0 0 0 1 0 0 1;
    0 0 0 0 1 0 0;
    0 0 0 0 1 0 0;
];
n = size(adjMatrix, 1); % Number of vertices
degrees = sum(adjMatrix, 2); % Degree of each vertex

% Step 1: Find support vertices on the spine (degree > 2)
support_vertices_on_the_spine = find(degrees > 2);
k = length(support_vertices_on_the_spine);
```

```
% Initialize color classes: one for each support vertex + 2 more
color_classes = cell(1, k + 2);
for i = 1:k
    color_classes{i} = support_vertices_on_the_spine(i);
end

% Step 2: Find remaining vertices
residual_vertices = setdiff(1:n, support_vertices_on_the_spine);
original_residual = residual_vertices;

% Step 3: Partition remaining vertices into 2 independent sets
color_class_1 = [];
color_class_2 = [];

if ~isempty(residual_vertices)
    u = residual_vertices(1);
    color_class_1 = [color_class_1, u];
    residual_vertices = setdiff(residual_vertices, u);

    for i = 1:length(residual_vertices)
        v = residual_vertices(i);
        v_neighbors = find(adjMatrix(v, :) == 1);

        conflict = false;
        for j = 1:length(v_neighbors)
            if any(color_class_1 == v_neighbors(j))
                conflict = true;
                break;
            end
        end

        if ~conflict
            color_class_1 = [color_class_1, v];
        else
            color_class_2 = [color_class_2, v];
        end
    end
```

```
end

color_classes{k + 1} = color_class_1;
color_classes{k + 2} = color_class_2;

% Step 4: If any P2 exists (edge between residual vertices)
P2_exists = false;
for i = 1:length(original_residual)
    for j = i + 1:length(original_residual)
        if adjMatrix(original_residual(i), original_residual(j)) == 1
            P2_exists = true;
            break;
        end
    end
    if P2_exists
        break;
    end
end

% Step 5: Calculate chromatic number
if P2_exists
    chi_pd = k + 2;
else
    chi_pd = k + 1;
end

% Output results
fprintf('Power Dominator Chromatic Number chi_pd(G) = %d\n', chi_pd);

for i = 1:length(color_classes)
    fprintf('Color Class C^(%d): ', i);
    disp(color_classes{i});
end
```

Output:

```
Power Dominator Chromatic Number chi_pd(G) = 3
Color Class C^(1): 2
Color Class C^(2): 1   3   4   6   7
Color Class C^(3): 5
```

8. Power Dominator Coloring in the Chennai Metro Feeder Network

The Chennai Metro Rail system, comprising key corridors such as the Blue and Green Lines, plays a vital role in the city's transportation infrastructure. Ensuring the reliability, safety, and efficiency of its operations demands a robust monitoring framework, particularly for its feeder networks, which are responsible for power distribution and data flow across various metro stations and associated nodes. Traditional sensor deployment across every node or substation leads to significant costs and logistical complexity. To address this, we apply the Power Dominator Coloring (PDC) algorithm, a graph-theoretic approach designed to minimize the number of monitoring units needed while guaranteeing full network observability.

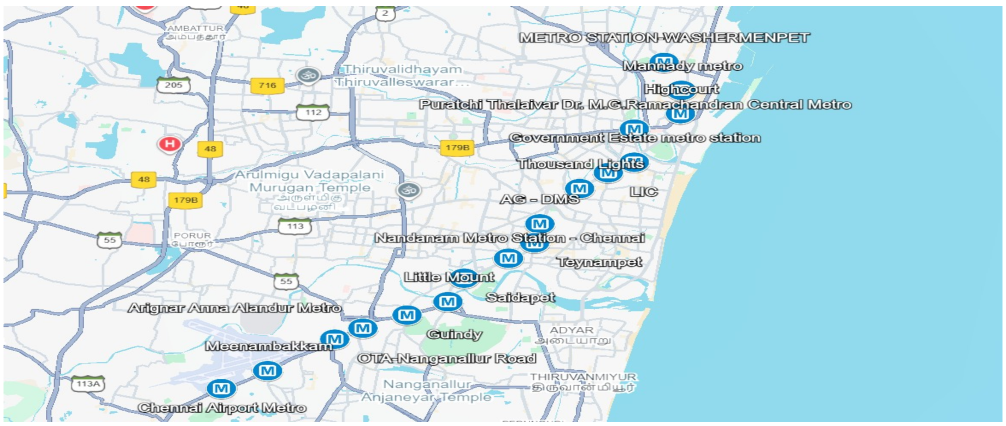

Figure 3. Chennai Metro map highlighting the operational Blue Line across the city. [Google Maps Chennai, Tamil Nadu. Retrieved 30 July, 2025, from https://maps.app.goo.gl/gQm3sKdoKu89VFgP6. Metro lines manually drawn by authors]

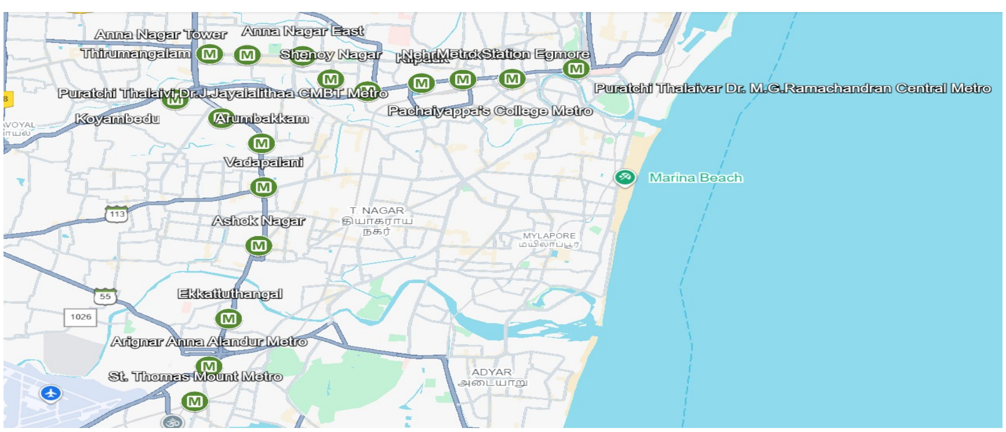

Figure 4. Chennai Metro map highlighting the operational Green Line across the city.[Google Maps Puratchi Thalaivar Dr. M.G.Ramachandran Central Metro. Retrieved 30 July, 2025, from https://maps.app.goo.gl/7jRZHHkMPSQbwVoL9. Metro lines manually drawn by authors]

Both the Blue Line (Washermanpet to Chennai International Airport) and the Green Line (Central Metro to St. Thomas Mount) can be modeled as caterpillar graphs, where spine vertices represent major metro stations and leaf vertices represent feeder connections such as bus stops or local substations. Using the PDC algorithm, we identify Support Vertices on the spine (SVs): stations with a degree greater than two, assigning each a unique color class. The remaining vertices are divided into two independent sets and are assigned one or two additional color classes depending on the presence of internal paths of length two. The computational process involves identifying high-degree vertices, partitioning sets using adjacency checks, and verifying path independence, all within a worst-case time complexity of $O(n^2)$, though typically operating in $O(n)$ time for smaller caterpillar graphs.

For the Blue Line, stations like Alandur, AG-DMS, and High Court are designated as SVs. They are each assigned a unique colors 1, 2 and 3 respectively (Color classes C^1 to C^3), while for other stations colored with additional colors 4 and 5 alternatively like Saidapet and Nandanam are grouped into additional color classes C^4 and C^5, resulting in a total of five color classes, as shown in Table 1.

For the Green Line, Central Metro, Egmore, and St. Thomas Mount considered as SVs and assigned unique colors 1, 2 and 3 respectively (assigned to Color classes C^1, C^2 and C^3 respectively), while for other stations like Ashok Nagar and vadapalani assigned to C^4 and C^5, again yielding five total color classes, clearly shown in Table 2. This structured coloring directly translates to optimized monitoring zones, where each dominator node supervises its associated color class. These dominator nodes are ideal locations for placing edge servers and IoT hubs, reducing infrastructure costs, and enabling localized emergency response zones.

Table 1. Functional Role Assignment of Stations on the Chennai Metro Blue Line Based on Power Dominator Coloring

Station	**Deg.**	**Role**	**Color Class**	**Function**
Alandur	4	SV	C^1	Surveillance, IoT
AG-DMS	4	SV	C^2	Emergency Control
High Court	4	SV	C^3	Monitoring
Chennai Airport	1	Terminal	C^5	Load Balancing & Security Check
Meenambakkam	2	Spine Vertex	C^4	Backup Power Transfer Node
OTA–Nanganallur Road	2	Spine Vertex	C^5	Passenger Access Monitoring
Guindy	2	Spine Vertex	C^4	Smart Transit Interface
Little Mount	2	Spine Vertex	C^5	Transit Coordination
Saidapet	2	Spine Vertex	C^4	Service Monitoring
Nandanam	2	Spine Vertex	C^5	Local Traffic Management
Teynampet	2	Spine Vertex	C^4	Emergency Alert Relay
Thousand Lights	2	Spine Vertex	C^5	Crowd Control / Alerts
LIC	2	Spine Vertex	C^4	Passenger Info Relay
Government Estate	2	Spine Vertex	C^5	Passenger Guidance
Puratchi Thalaivar Dr. M.G.R. Central Metro	3	Interchange Hub	C^4	High-Speed Data Hub
Mannadi	2	Spine Vertex	C^5	Feeder Optimization
Washermenpet	1	Terminal	C^4	Last-mile Connectivity

Table 2. Functional Role Assignment of Stations on the Chennai Metro Green Line Based on Power Dominator Coloring

Station	Degree	Role	Color Class	Function
Central Metro	5	SV	C^1	Core Hub
Egmore	4	SV	C^2	Data Aggregation
St. Thomas Mount	4	SV	C^3	Terminal with Feeders
Alandur Metro	3	Interchange Hub	C^4	Interchange Controller
Ekkattuthangal	2	Spine Vertex	C^5	IoT Sensor Aggregator
Ashok Nagar	2	Spine Vertex	C^4	Zone Load Management
Vadapalani	2	Spine Vertex	C^5	Power Redundancy Monitoring
Arumbakkam	2	Spine Vertex	C^4	Crowd Detection System
CMBT	3	Spine Vertex	C^5	Data Router for Bus & Metro Hub
Koyambedu	2	Spine Vertex	C^4	Emergency Power Control
Thirumangalam	2	Spine Vertex	C^5	Incident Detection Node
Anna Nagar Tower	2	Spine Vertex	C^4	Feeder Demand Balancing
Anna Nagar East	2	Spine Vertex	C^5	Zonal Surveillance
Shenoy Nagar	2	Spine Vertex	C^4	Backup Data Routing
Pachaiyappa's College	2	Spine Vertex	C^5	Crowd Flow Optimization
Kilpauk	2	Spine Vertex	C^4	Transit Coordination
Nehru Park	2	Spine Vertex	C^5	Local Energy Management

In a real sense, the use of PDC has very significant consequences. Above all, it significantly reduces the amount of sensors and monitoring apparatus required, reducing deployment and maintenance expenses. Improve network resilience by ensuring that any anomaly is quickly detected and localized to one color class. Third, it allows for smart energy and load control through the means of zone-by-zone ventilation, lighting,and other energy-intensive services during off-peak hours. Furthermore, all traffic data-whether passenger pedestrian counters or ambient sensors-is forwarded by dominator nodes, enhancing data aggregation and processing efficiency.

The same PDC strategy was used to optimize the electrical feeder network that serves the metro corridors. Substations, converters, and power lines as vertices and edges in a caterpillar-like graph. The PDC algorithm helped in determining the minimum number of substations were required to supply power at maximum capacity without voltage drops or redundancies. The simulation outcomes indicated improved voltage profiles, streamlined scheduling for maintenance, and fault tolerance.

Although effective, the PDC model does pose practical issues. Real-world feeder networks are often more complex than caterpillar graphs, demanding graph representation approximation techniques. Accurate, up-to-Date topology information is also required for effective implementation. Scalability takes on significance with the growth of the metro system and cybersecurity being an priority to protect the communication and control systems linked to dominator nodes.

Because, as it follows from the conclusion, the real conditions can differ from the idealized caterpillar graph structure model used in the research. Future studies must develop dynamic implementations of the Power Dominator Coloring algorithm

9. Conclusion

This article analysed the Power Dominator Coloring (PDC) of caterpillar graphs with structural constraints and an algorithm to compute the power dominator chromatic number χ_{pd} of a caterpillar graph. We established that for every caterpillar G having t support vertices on the spine, the chromatic number has the bound $2 \leq \chi_{pd}(G) \leq t + 2$, and established that the bound is tight. A MATLAB implementation has been provided to construct optimal colorings with reference to structural properties such as spine and support vertex distribution. An example case study of the Chennai Metro feeder network that linked Guindy with Washermanpet proved the feasibility of the algorithm. The case study showed that support vertices on the spine can be utilized as central monitoring nodes, limiting the number of resources required for complete observability. The results justify the usefulness of PDC in infrastructure development, particularly for networked transport and metropolitan monitoring systems. Subsequent research can diverge from here methodology to dynamic graph structures and more sophisticated ones like grids and hybrid urban networks.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] T.W. Haynes, S.T. Hedetniemi, M.A. Henning, Domination in Graphs: Core Concepts, Springer, 2023. https://doi.org/10.1007/978-3-031-09496-5.

[2] R.M. Gera, On Dominator Colorings in Graphs, in: Graph Theory Notes of New York, New York Academy of Sciences, LII, 2007, pp. 25–30.

[3] F. Harary, Graph Theory, Addison-Wesley Publishing Co., Reading, 1969.

[4] T.W. Haynes, S.M. Hedetniemi, S.T. Hedetniemi, M.A. Henning, Domination in Graphs Applied to Electric Power Networks, SIAM J. Discret. Math. 15 (2002), 519–529. https://doi.org/10.1137/S0895480100375831.

[5] M. Dorfling, M.A. Henning, A Note on Power Domination in Grid Graphs, Discret. Appl. Math. 154 (2006), 1023–1027. https://doi.org/10.1016/j.dam.2005.08.006.

[6] K. Koh, K. Soh, On the Power Domination Number of the Cartesian Product of Graphs, AKCE Int. J. Graphs Comb. 16 (2019), 253–257. https://doi.org/10.1016/j.akcej.2019.02.004.

[7] A.U. Maheswari, B. Samuvel, Power Dominator Chromatic Number for Some Special Graphs, Int. J. Innov. Technol. Explor. Eng. 8 (2019), 3957–3960. https://doi.org/10.35940/ijitee.L3466.1081219.

[8] M. Zhao, L. Kang, G.J. Chang, Power Domination in Graphs, Discret. Math. 306 (2006), 1812–1816. https://doi.org/10.1016/j.disc.2006.03.037.

[9] M. Chellali, F. Maffray, Dominator Colorings in Some Classes of Graphs, Graphs Comb. 28 (2011), 97–107. https://doi.org/10.1007/s00373-010-1012-z.

[10] M. Shukla, F. Chandarana, Dominator Coloring of Total Graph of Path and Cycle, Math. Model. Eng. 9 (2023), 72–80. https://doi.org/10.21595/mme.2023.23228.

[11] I. Chandramani, A.S.P. Venkatesan, S. Sriram, Power Dominator Chromatic Numbers of Jahangir and Associated Graph, Adv. Appl. Math. Sci. 21 (2022), 6351–6359.

[12] K.S. Kumar, N.G. David, K.G. Subramanian, Graphs and Power Dominator Colorings, Ann. Pure Appl. Math. 11 (2016), 67–71.

[13] D.B. West, Introduction to Graph Theory, Prentice Hall, Upper Saddle River, 1996.

[14] L.W. Beineke, R.E. Pippert, The Number of Labeled *k*-Dimensional Trees, J. Comb. Theory 6 (1969), 200–205. https://doi.org/10.1016/S0021-9800(69)80120-1.