Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 4, pp. 1777–1786
DOI : 10.47974/JDMSC-2573

Edge forcing in tree-like architectures

G. Jessy Sujana [†]
Department of Computer Science and Engineering (Emerging Technologies)
SRM Institute of Science and Technology
Vadaplani
Chennai 600026
Tamil Nadu
India

J. Anitha *
Department of Mathematics
Rajalakshmi Institute of Technology
Chennai 600124
Tamil Nadu
India

T. M. Rajalaxmi [§]
Department of Mathematics
Sri Sivasubramaniya Nadar College of Engineering
Chennai 603110
Tamil Nadu
India

Abstract

To create independent edges in a graph G, this work presents a variant of the zero forcing set called the 'edge-forcing set.' A collection of vertices in G that can influence their neighbors to force all vertices to become forced iteratively is called a zero forcing set. If a forced vertex has exactly one unforced neighbor at any given step, it forces that neighbor to become forced as well. This concept is extended to edges by the edge-forcing set. We show that figuring out the edge-forcing number, the smallest number of vertices required to build an edge-forcing set is an NP-complete task. Additionally, for a variety of tree-like structures,

[†] *E-mail:* jessysug@srmist.edu.in
* *E-mail:* anithaharish78@gmail.com (Corresponding Author)
[§] *E-mail:* laxmi.raji18@gmail.com

© Taru Publications

we investigate the edge-forcing number, giving precise values for sibling trees and defining bounds for hypertree.

Subject Classification (2020): *05C69, 05C50.*

Keywords: *Zero forcing set, Forced vertex, NP-complete task, Edge forcing number, Unforced neighbor.*

1. Introduction

A graph's propagation model is analogous to an activation process, in which a starting set of active vertices iteratively applies a given activation rule to gradually activate the remaining vertices. Until no more vertices can be activated, this process is repeated. The goal of this research is to determine, using a certain forcing technique, the minimal size of such a collection of active vertices.

[14]-[19] explore advanced fuzzy graph theory and its applications in multi-attribute decision-making, psychological sciences, and medicine. The works by S. Kosari and collaborators delve into various types of fuzzy and vague graphs, including q-rung orthopair fuzzy Zagreb indices, perfectly regular fuzzy graphs, and vague domination concepts. These models are applied to real-world scenarios such as decision-making problems and psychological analysis, offering new mathematical frameworks for uncertainty handling. Collectively, the studies contribute to the growing field of soft computing, offering new perspectives and methodologies that bridge theoretical graph structures with practical problem-solving across disciplines.

First proposed in relation to the minimal rank problem, the 'zero forcing number' is a graph parameter [3]. In 2007, [4] and [5] introduced the idea of a zero forcing set, sometimes referred to as a" infecting set," in relation to quantum systems. It has been demonstrated that determining the zero forcing number is NP-complete [6] and has been used in theoretical computer science, especially as a mixed search model [2].

In the zero forcing problem, the objective is to find the smallest set of nodes S in a graph G such that all other nodes can be activated through the forcing process [1].

"Total forcing" [12, 7], "Connected forcing" [8, 9, 10], and "k-forcing" [11, 4] are some of the forcing variations that have been investigated. The whole force problem has also been demonstrated to be NP-complete [12]. The "Edge-Forcing Problem" is a new problem introduced in this paper that has been shown to be NP-complete [13]. For particular tree-like architectures, we also investigate the edge-forcing number, obtaining precise values. In order to better comprehend structural features and optimise complex networked systems, recent research has examined the structural and domination-related properties of graph networks as well as methods pertaining to k-power domination problems [20, 21, 22, 23].

2 Main Result

2.1 *Hypertree*

An interconnection structure for computer systems with several processors that may be expanded progressively is called a hypertree. It blends the characteristics of a hypercube and a binary tree. It has the easily expandable feature of tree structures and is small, much like a hypercube. When implementing multiprocessor networks, the hypertree structure is appealing. As demonstrated by Rajasingh et al., Figure 1 is isomorphic.

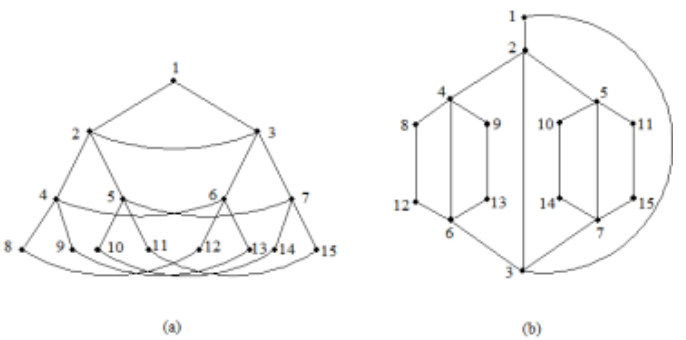

Figure 1
(a) Original drawing of *HT*(3) and (b) *HT*(3) redrawn

2.2 *Edge-forcing in Hypertree*

Definition 2.1: Let *HT**(2) be the graph obtained from HT(2) by deleting the root vertex of degree 2. Let the two children of the root vertex of *HT*(2) be denoted by *u* and *v* as shown in Figure 2.

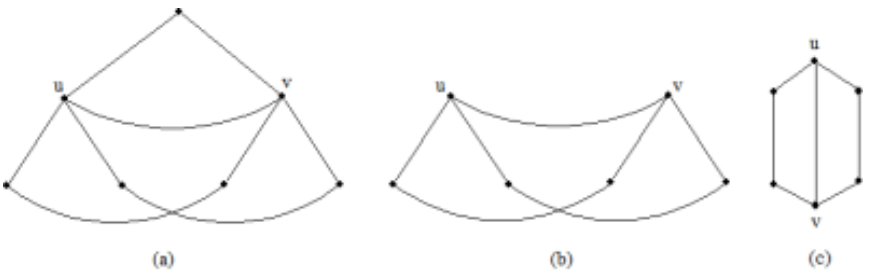

Figure 2
(a) *HT*(2) (b) *HT(2) (c) *HT**(2) redrawn**

Lemma 1: *Let HT*(2) be an induced subgraph of a graph G with u and v as cut vertices and for* $x \in V(HT^*(2)) \{u, v\}$, $\deg_G x = \deg_{HT^*(2)} x$. *Then an edge-forcing set of G contains at least one edge of HT*(2).*

Proof: Let's say not. The remaining vertices of *HT**(2) cannot be monitored even if u and v are monitored by edges that are not in *HT**(2) because the

monitoring procedure cannot be carried out with both u and v next to two unmonitored vertices of $HT^*(2)$, respectively. This suggests that there is at least one edge in $HT^*(2)$ in any edge-forcing set of G. Remark 1: As induced subgraphs, we see that $HT(r)$, $r \geq 2$ contains 2^{r-2}copies of $HT^*(2)$, each of which satisfies the requirements of Lemma 1. Thus, we derive the following Lemma from Lemma 1, $r \geq 2$, and the structure of $HT(r)$.

Lemma 2: $\zeta_e(HT(n)) \geq 2r\text{-}2$, $r \geq 2$. *We now proceed to obtain an upper bound for* $\zeta_e(HT(n))$, $r \geq 2$.

Analogous to $HT^*(2)$, we define $HT^*(i)$ as the graph obtained from $HT(i)$, by deleting the root node of $HT(i)$, $3 \leq i \leq r$. See Figure 3.

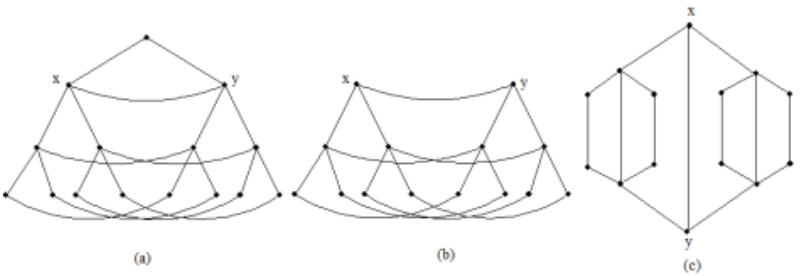

Figure 3
(a) *HT*(3) (b) *HT(3) (c) *HT**(3) redrawn**

Remark 2: 2^{r-3}copies of $HT^*(3)$ are present as induced subgraphs in $HT(r)$, $r \geq 3$.

Lemma 3: $\zeta_e(HT(n)) \leq 3.\ 2^{r-3}$, $r \geq 3$.

Proof: As illustrated in Figure 4, select three edges from a subgraph $HT^*(3)$ of $HT(r)$.

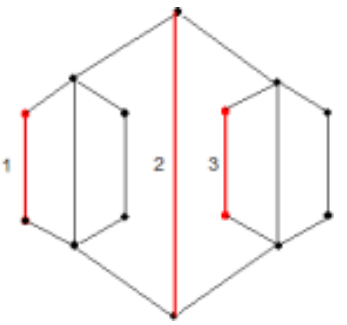

Figure 4
***HT**(3) with chosen edges marked in red**

Similarly, in $HT(r)$, $r \geq 3$, choose three of these edges in each of the 2^{r-3}copies of $HT^*(3)$. Refer to Figure 5.

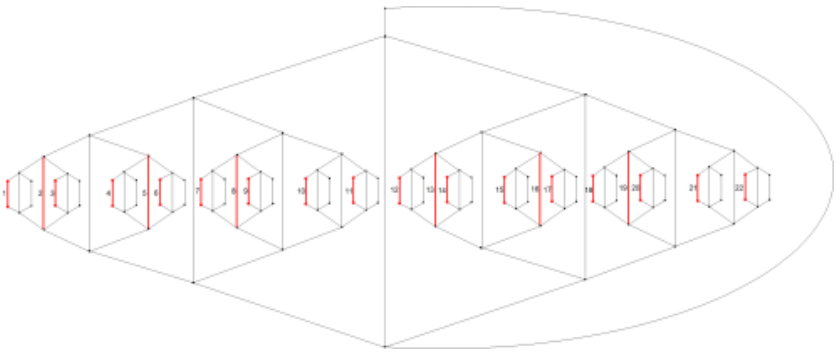

Figure 5
***HT*(6) with forcing edges marked in red**

By using induction on the dimension r of $HT(r)$, $r \geq 3$, we demonstrate the result. Think about $HT(3)$. It is possible to manually verify that every vertex in $HT^*(3)$ is being watched from Figure 4. This suggests that the same set of edges is watching over $HT(3)$. Refer to Figure 6.

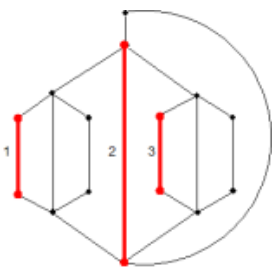

Figure 6
Edge-forcing set of *HT*(3)

Assume that once the edges are chosen, 3.2^{r-4} edges continue to monitor $HT(r\text{-}1)$ and $HT^*(r\text{-}1)$. $HT(r\text{-}1)$ contains exactly two copies of $HT^*(r\text{-}1)$. Therefore, all the vertices of $HT(r)$ are monitored by 3.2^{r-3} edges of $HT(r)$. Consequently, the following outcome is obtained by $\zeta_e(HT(n)) \leq 3.2^{r-3}$, $r \geq 3$. By Lemma 2 and Lemma 3,

Theorem 1: *Let HT(r) be the hypertree of dimension r, r ≥ 3. Then,* $2^{r-2}\zeta_e(HT(r)) \leq 3/2.\ 2^{r-2}$, $r \geq 3$.

2.3 X-Tree

The *X*-tree is a variation of the *R*-tree, a tree-based data structure designed for indexing multi-dimensional data such as bounding boxes, rectangles, and geographic information. *X*-trees are specifically built to handle high-dimensional feature spaces by reducing overlap and redundancy.

Definition 2.2: A complete binary tree on 2^{r+1} - 1 vertices yield a X-tree XT(r) by adding pathways P_i of length 2^{i-1} across all of the vertices at level i, from left to right. Remark 3: In the X-tree XT(r), r ≥ 3, the vertices in pairings of levels i and i - 1 cause i = r, r - 2, r - 4,... 2 or 1 depending on whether r is even or odd. This results in 2^{r-2} number of vertex disjoint copies of K_3.

2.4 *Edge-Forcing in X-Tree*

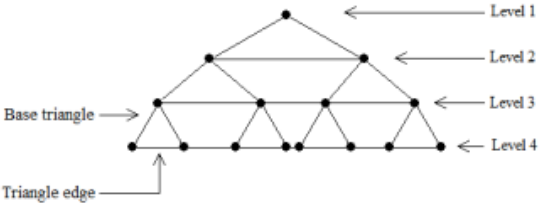

Figure 7
***XT*(4) with vertices in 4 levels**

Level *i* in Figure 7 has 2*i*-1 vertices for $1 \leq i \leq r$. An edge in Level *r* is referred to as a *triangle edge* if it induces a 3-cycle when its end vertices are combined with a vertex in Level (*r*-1). A 3-cycle with an apex vertex in Level (*r*-1) and one edge in Level *r* is referred to as a *base triangle*. *XT*(*r*) contains 2^{r-2} base triangles, $r \geq 2$.

Theorem 2: $\zeta_e(HT(r)) \leq 2r\text{-}1$.

Proof: An edge-forcing set of *XT*(*r*) is the collection of triangular edges from each of the basis triangles of *XT*(*r*). Starting from the leftmost vertex in Level *r*, these edges keep an eye on every vertex in Level (*r*-1), starting with the leftmost vertex in Level (*r*-1). Subsequently, the monitored vertices in Level *i*, $r\text{-}1 \geq i \geq 1$, monitor every vertex in Level (*i*-1). $\zeta_e(HT(r)) \leq 2r\text{-}1$, because of this. Refer to Figure 7-8.

Conjecture: $\zeta_e(HT(r)) = 2r\text{-}1$.

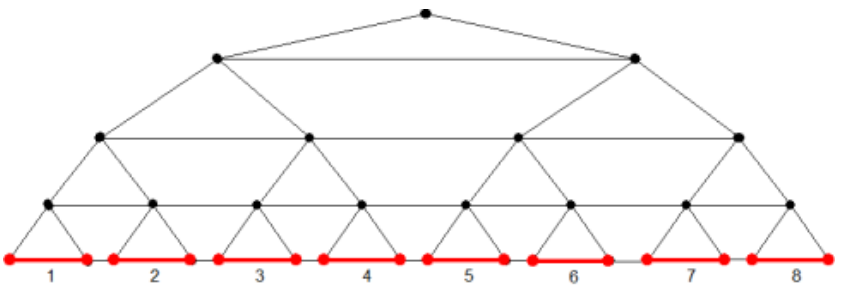

Figure 8
***XT*(4) with triangle edges (forcing edges) marked in red**

2.5 *Sibling Tree*

A tree data structure having nodes related to their siblings—that is, nodes with the same parent—is called a sibling tree. Sibling trees are appropriate for depicting more intricate hierarchical relationships because, in contrast to binary trees, they are not constrained to a set number of offspring.

2.6 *Edge-Forcing in Sibling Tree*

As illustrated in Figure 9, Level i has $2i$ vertices for $0 \leq i \leq r$. In Level r, an edge is referred to as a triangle edge, and the 3-cycle that contains it is known as a base triangle. $SibT(r)$ has $2r$-1 base triangles.

Lemma 4: *An edge-forcing set of SibT(r), $r \geq 1$, possesses at least one edge from each base triangle.*

Proof: Let's say not. Let T denote a base triangle in an edge-forcing set of $SibT(r)$ that initially has no edges. Let v represent T's apex vertex in Level (r-1). Since V is next to two unmonitored vertices of T, the monitoring process cannot proceed even if v is monitored. This demonstrates that at least one edge of each base triangle is present in the edge-forcing set of $SibT(r)$. Because we have $2r$-1 base triangles in $SibT(r)$, $r \geq 1$, we thus arrive at the following conclusion. Lemma 5: $\zeta_e(HT(r)) \geq 2r$-1.

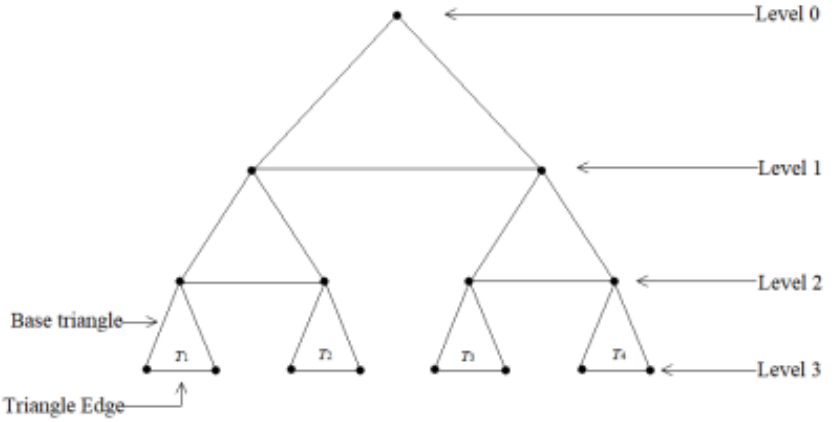

Figure 9
***SibT*(3) with base triangles T_1, T_2, T_3, T_4**

Theorem 3: $\zeta_e(HT(r)) = 2r$-1, $r \geq 1$.

Proof: Let F be the collection of all $SibT(r)$ triangle edges. $|F|$ then equals $2r$-1. These edges keep an eye on every vertex in Level (r-1), which in turn keeps an eye on every vertex in the remaining levels. $\zeta_e(HT(r)) = 2r$-1, $r \geq 1$. Refer to Figure 10.

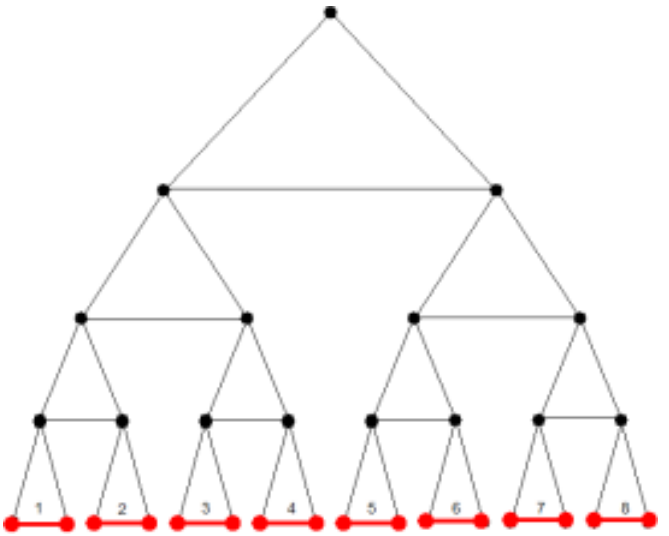

Figure 10
***SibT*(4) with triangle edges (forcing edges) marked in red**

3. Conclusion

In tree-like designs such as Hypertrees, *X*-trees, and Sibling trees, the edge-forcing set has been investigated. Bounds for Hypertrees and *X*-trees have been defined, while the precise edge-forcing number for Sibling trees has been established.

References

[1] K. F. Benson, D. Ferrero, M. Flagg, V. Furst, L. Hogben, V. Vasilevska, and B. Wissman, "Power domination and zero forcing", arXiv:1510.02421 (2018).

[2] D. Ferrero, C. Grigorious, T. Kalinowski, J. Ryan, and S. Stephen, "Minimum rank and zero forcing number for butterfly networks", *Journal of Combinatorial Optimization*, vol. 37, pp. 970–988 (2019).

[3] AIM Minimum Rank - Special Graphs Work Group, "Zero forcing sets and the minimum rank of graphs", *Linear Algebra, and its Applications*, vol. 428, pp. 1628-1648 (2008).

[4] D. Burgarth and V. Giovannetti, "Full control by locally induced relaxation", *Physical Review Letters*, vol. 99, pp. 1-7 (2007).

[5] D. Burgarth and K. Maruyama, "Indirect Hamiltonian identification through a small gateway", *New Journal of Physics*, vol. 11, no. 103019, pp. 1-9 (2009).

[6] S. Fallat, K. Meagher, and B. Yang, "On the complexity of the positive semidefinite zero forcing number", *Linear Algebra, and its Applications*, vol. 491, pp. 101-122 (2016).

[7] R. Davila and M. A. Henning, "On the total forcing number of a graph", *Discrete Applied Mathematics,* vol. 257, pp. 115-127 (2019).

[8] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetneimi, and M. A. Henning, "Domination in graphs applied to electric power networks", *SIAM Journal on Discrete Mathematics*, vol. 15, pp. 519- 529 (2002).

[9] R. Davila, M. A. Henning, C. Magnant, and R. Pepper, "Bounds on the Connected Forcing Number of a Graph", *Graphs and Combinatorics*, vol. 34, pp. 1159–1174 (2018).

[10] D. Ferrero, S. Varghese, and A. Vijayakumar, "Power domination in honeycomb networks", *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 14, pp. 521-529 (2011).

[11] D. Amos, Y. Caro, R. Davila, and R. Pepper, "Upper bounds on the k-forcing number of a graph", *Discrete Applied Mathematics,* vol. 180, pp. 1-10 (2015).

[12] R. Davila and M. A. Henning, "On the Total Forcing Number of a Graph", *Discrete Applied Mathematics*, vol. 257, pp. 115-127 (2019).

[13] J. Sujana, T. M. Rajalaxmi, S. Rajan, and I. Rajasingh, "Edge Forcing in Butterfly Networks", *Fundamenta Informaticae*, vol. 182, no. 3, pp. 285-299 (2021).

[14] Y. Rao, S. Kosari, S. Hameed, and Z. Yousaf, "Multi-attribute decision-making using q-rung orthopair fuzzy Zagreb index", *Artificial Intelligence Review*, vol. 58, no. 153, pp. 1-31 (2025).

[15] S. Kosari, X. Qiang, J. Kacprztk, Q. Ain, and H. Rashmanlou, "A Study on Topological Indices in Fuzzy Graphs with Application in Decision Making Problems", *Journal of Multiple-Valued Logic & Soft Computing,* vol. 42, no. 5-6, pp. 567-589 (2024).

[16] S. Kosari, X. Shi, J. Kacprztk, Z. Chen, and H. Rashmanlou, "A Novel Description of Perfectly Regular Fuzzy Graphs with Application in Psychological Sciences", *Journal of Multiple-Valued Logic & Soft Computing*, vol. 42, no. 5-6, pp. 405-412 (2024).

[17] X. Shi, S. Kosari, P. Rangasamy, R. K. Nivedhaa, and H. Rashmanlou, "Scaled aggregation operations over three-dimensional extended intuitionistic fuzzy index matrices". *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, vol. 48, no. 1-2, pp. 123-139 (2025).

[18] S. Kosari, Z. Shao, Y. Rao, X. Liu, R. Cai, and H. Rashmanlou, "Some Types of Domination in Vague Graphs with Application in Medicine", *Journal of Multiple-Valued Logic & Soft Computing*, vol. 40, no. 3-4, pp. 203-219 (2023).

[19] Z. Shao, S. Kosari, Y. Rao, H. Rashmanlou, and F. Mofidnakhaei, "New Kind of Vague Graphs with Novel Application", *Journal of Multiple-Valued Logic & Soft Computing*, vol. 40 (2023).

[20] G. K. Nandini, S. Klavžar, T. M. Rajalaxmi, and R. S. Rajan, "A note on eccentricity based topological indices of honeycomb, oxide and 2-power interconnection networks", *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 26, no. 1, pp. 231–253 (2023).

[21] S. Arulanand, R. S. Rajan, S. Prabhu, and S. Stephen, "Certain domination numbers for Cartesian product of graphs", *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 3, pp. 1045–1058 (2024).

[22] G. M. Jose, K. N. Geetha, and K. Somasundaram, "Double power domination in graphs", *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 28, no. 6, pp. 2261–2278 (2025).

[23] B. S. Mahdi, J. H. H. Al-Bayati, and A. J. Al-Mahdawi, "Minimum spanning tree application in Covid-19 network structure analysis in the countries of the Middle East", *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 8, pp. 2723-2728 (2022).

Received May, 2025