Fault-Tolerant Basis of Generalized Fat Trees and Perfect Binary Tree Derived Architectures

S. Prabhu[a,*], V. Manimozhi[b], Akbar Davoodi[c], Juan Luis García Guirao[d]

[a]Department of Mathematics, Rajalakshmi Engineering College, Thandalam, Chennai 602105, India

[b]Department of Mathematics, Panimalar Engineering College, Chennai 600123, India

[c]The Czech Academy of Sciences, Institute of Computer Science

Pod Vodárenskou věží 2, 182 07 Prague, Czech Republic

[d]Department of Applied Mathematics and Statistics, Technical University of Cartagena, Cartagena, Spain

Abstract

The ability to uniquely identify all nodes in a network based on network distances has proven to be highly beneficial despite the computational challenges involved in discovering minimal resolving sets within an interconnection network. A subset R of vertices of a graph G is referred to as a resolving set of the graph if each node can be uniquely identified by its distance code with respect to R, with its minimal cardinality defining the metric dimension of G. Similarly, a resolving set $F \subseteq V$ is designated as a fault-tolerant resolving set if $F \setminus \{s\}$ serves as a resolving set for each $s \in F$. The minimum cardinality of F represents the fault-tolerant metric dimension of G. Although determining the precise metric dimension of a given graph remains challenging, various effective techniques and meaningful constraints have been developed for different graph families. However, no notable technique has been developed to find fault-tolerant metric dimension of a given graph. Recently, Prabhu et al. have shown that each twin vertex of G belongs to every fault-tolerant resolving set of G. Consequently, the fault-tolerant metric dimension is equal to the order of the graph G if and only if each vertex of G is a twin vertex, a characterization proved in [Appl. Math. Comput. **420** (2022) 126897] corrects a wrong characterization in the literature. It is also interesting to note from the above literature correction that the twin vertices are necessary to form the fault-tolerant resolving set, but determining whether they are sufficient is challenging. Evidence of this context is also discussed in this paper through the amalgamation of perfect binary trees. This article focuses on determining the exact value of the fault-tolerant metric dimension of generalized fat trees. For the amalgamation of perfect binary trees, both the metric dimension and fault-tolerant metric dimension were precisely found.

Keywords: Basis; Fault-tolerant basis; Twin vertices; Generalized fat tree; Perfect binary trees

*Corresponding author: drsavariprabhu@gmail.com

Mathematics Subject Classification (2010): 05C12, 68M15

1 Introduction

Networks can be used to model various systems and phenomena, such as social networks, transportation systems, computer networks, biological networks, and many others. Graph theory provides tools and techniques for analyzing and understanding the structure and properties of these networks. Network analysis involves studying various properties of networks, such as connectivity, centrality, clustering, and resilience, among others. This analysis helps in understanding the structure, behaviour, and functionality of complex systems represented by networks.

Interconnection networks are crucial components in parallel and distributed computing systems. They provide the infrastructure for communication among different processing elements, such as CPUs, memory units, and I/O devices. These networks play a vital role in achieving high performance and scalability in large-scale computing systems. Interconnection networks are fundamental in building parallel and distributed computing systems, including supercomputers, data centers, and high-performance computing clusters. Designing efficient and scalable interconnection networks is essential for achieving optimal performance and scalability in these systems.

Network verification, as considered in [1], aims to determine the minimum number of queries required to verify all edges and non-edges of a graph with a given network. A process of verification is undertaken to identify all edges and non-edges with endpoints at distinct distances from the query vertex v. This problem has previously been addressed under various names, such as landmark placement in a graph or establishing the basis of a graph [2]. According to graph-theoretic principles, the solution involves providing specific vertex representations for each vertex in the graph, a concept extensively discussed in the literature [3–5].

The preliminaries, along with all notation used in this manuscript, definitions and figures which help to understand the definition are given in Section 2. Also, in the same section, the problem and its overview are presented along with the running example. In subsequent sections, we delve into the two families of network structures: generalized fat trees (Section 3.1) and amalgamations of perfect binary trees (Section 3.2). We continue to discuss the metric dimension and FTMD for two families in Section 4, especially the Lemma 7, which consults that the twin vertices necessary to form a basis need not be sufficient. The discussions are supplemented with related results and concluding remarks. The concluding remarks discussed an exciting application of metric dimension and fault-tolerant metric dimension. Finally, the conclusion concludes with a challenging open problem for the future researcher.

2 Preliminaries, the Problem and its Overview

Throughout the paper, all graphs are assumed to be undirected and simple. To be mathematically precise, let us consider a connected graph $G = (V, E)$ with vertex set $V(G)$ and edge set $E(G)$. For any two vertices

$s, t \in V(G)$, the distance $d(s,t)$ is defined as the number of edges in a minimum path that connects s and t. Given a vertex s in graph G, the open neighborhood of s, denoted as $N_G(s)$, collects all vertices t in $V(G)$ such that $st \in E(G)$. On the other hand, the closed neighborhood of s, denoted as $N_G[s]$, encompasses $N_G(s)$ as well as the vertex s itself. If $N_G[s] = N_G[t]$, then two distinct vertices s, t are referred to as adjacent twins, and when $N_G(s) = N_G(t)$, they are named as non-adjacent twins. If vertices s and t exist such that s and t are twins, then s is designated as a twin vertex for t. When each pair of vertices $s, t \in \mathcal{T}$ are twins, the set $\mathcal{T}$ is identified as a twin set of G. See Figure 1.

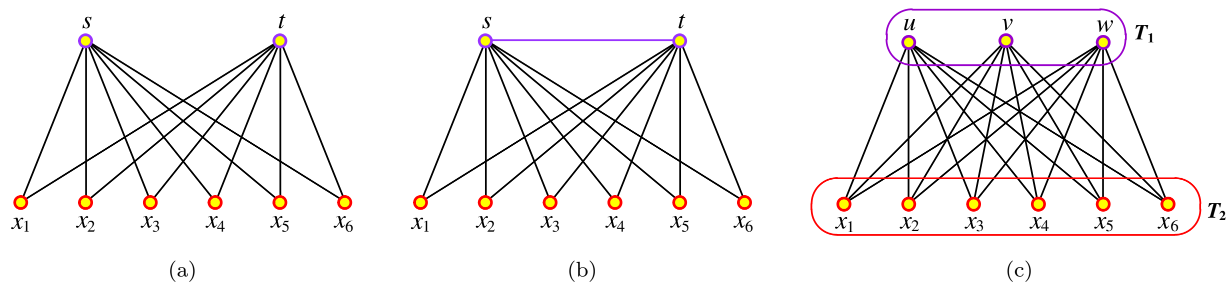


Figure 1: (a) Non-adjacent twins ($N_G(s) = N_G(t) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$); (b) Adjacent twins ($N_G[s] = N_G[t] = \{x_1, x_2, x_3, x_4, x_5, x_6, s, t\}$); (c) Twin sets T_1 and T_2 ($T_1 = N_G(x_i) = N_G(x_j) = \{u, v, w\}$ for every $x_i, x_j \in T_2$ and $T_2 = N_G(u) = N_G(v) = N_G(w) = \{x_1, x_2, x_3, x_4, x_5, x_6\}$)

Resolving sets offer a solution for locating the source of diffusion within a network. For instance, identifying the origin of a disease spreading across a population could prove invaluable in various scenarios. While resolving sets offer a direct solution when inter-node distances and the initial spread time are known, the concept of resolvability must be expanded to accommodate arbitrary start times and irregular nodal transmission delays.

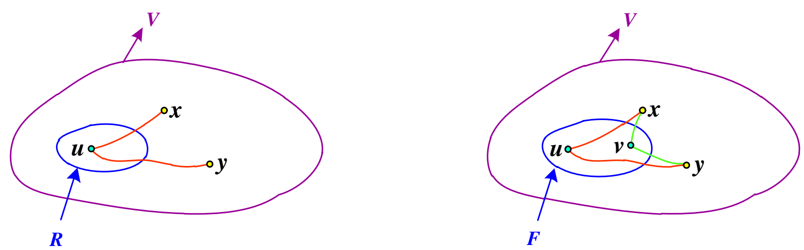
For a subset of vertices $R = \{r_1, r_2, \ldots, r_k\} \subseteq V(G)$, the *code* or *representation* of a vertex $s \in V(G)$ with respect to R is defined as the k-vector

$$C_R(s) = \Big(d(s, r_1), d(s, r_2), \ldots, d(s, r_k)\Big),$$

where $d(s,t)$ represents the distance between the vertices s and t. The set R is called a *resolving set* for G if, for every pair of distinct vertices x and y in $V(G)$, the codes $C_R(x)$ and $C_R(y)$ are distinct. In other words, R is a resolving set for G if, for every x and y in $V(G)$, there exists a vertex $u \in R$ such that $d(x,u) \neq d(y,u)$. Refer to Figure 2(a). Among all possible resolving sets for G, those with the minimum size are of interest, called basis. The cardinality of a minimum size resolving set is called the *metric dimension* of G, denoted by $\dim(G)$. Table 1 summarises the notations required and discussed in this paper.

Chartrand and Zhang introduced the idea of using basis elements as sensors [6]. In this approach, sensor failures due to defects can lead to the system's inability to detect events like intrusion or disruption. The concept of a fault-tolerant basis addresses this issue by presuming that a malfunctioning sensor won't cause overall system failure, as the other sensors can manage the intruder. Consequently, the fault-tolerant metric dimension exhibits a similar range of diversity as the classical metric dimension. For more insights on implementing fault-tolerance in resolvability and its mathematical properties, readers can refer to [7–9].

A fault-tolerant resolving set is defined as a subset of vertices F, where for every element $s \in F$, the set $F \setminus \{s\}$ functions as a resolving set for the graph G. In simpler terms, for any pair of distinct vertices $x, y \in V(G)$, a fault-tolerant resolving set ensures the existence of two vertices $u, v \in F$ such that $d(u, x) \neq d(u, y)$ and $d(v, x) \neq d(v, y)$. For a visual representation, please refer to Figure 2(b). The size of the smallest fault-tolerant resolving set or fault-tolerant basis is known as the Fault-Tolerant Metric Dimension (FTMD), denoted as $\dim'(G)$.



(a) (b)

Figure 2: (a) Resolving set R: Vertex u in R has distinct distances from vertices x and y, resolving them from each other. (b) Fault-tolerant resolving set F: Vertices u and v in F independently resolve vertices x and y.

The following Figure 3(a)-(c) respectively illustrates an example for resolving set, basis and fault-tolerant basis. In particular, Figure 3(a) depicts the resolving set $R_1 = \{v_5, v_6, v_3\}$, which is not a basis (since it is not a minimum resolving set). We claim this R_1 as a resolving set due to the distinct representations represented as follows: $C_{R_1}(v_1) = (3, 2, 2)$, $C_{R_1}(v_2) = (3, 2, 1)$, $C_{R_1}(v_7) = (2, 1, 2)$, and $C_{R_1}(v_4) = (1, 2, 1)$. Figure 3(b) depicts the minimum resolving set $R_2 = \{v_1, v_2\}$, the representation of other vertices with respect to R_2 are given by $C_{R_2}(v_3) = (2, 1)$, $C_{R_2}(v_4) = (3, 2)$, $C_{R_2}(v_5) = (3, 3)$, $C_{R_2}(v_6) = (2, 2)$, and $C_{R_2}(v_7) = (1, 2)$. Due to the argument in [2], which says that the $\dim(G) = 1$ if and only if G is a path, one cannot resolve this graph with less than two vertices as this graph is not isomorphic to the path. The last Figure 3(c) depicts the fault-tolerant basis $R_3 = \{v_1, v_2, v_7\}$. It is evident that the set $R_3 \setminus \{v_1\}$ gives the representation of the other vertices in the graphs as $C_{R_3 \setminus \{v_1\}}(v_3) = (1, 2)$, $C_{R_3 \setminus \{v_1\}}(v_4) = (2, 3)$, $C_{R_3 \setminus \{v_1\}}(v_5) = (3, 2)$, $C_{R_3 \setminus \{v_1\}}(v_6) = (2, 1)$, and $C_{R_3 \setminus \{v_1\}}(v_1) = (1, 1)$. Similarly, the set $R_3 \setminus \{v_2\}$ gives the representation as $C_{R_3 \setminus \{v_2\}}(v_3) = (2, 2)$, $C_{R_3 \setminus \{v_2\}}(v_4) = (3, 3)$, $C_{R_3 \setminus \{v_2\}}(v_5) = (3, 2)$, $C_{R_3 \setminus \{v_2\}}(v_6) = (2, 1)$, and $C_{R_3 \setminus \{v_2\}}(v_2) = (1, 2)$, and lastly, the set $R_3 \setminus \{v_7\}$ gives the representation as $C_{R_3 \setminus \{v_7\}}(v_3) = (2, 1)$, $C_{R_3 \setminus \{v_7\}}(v_4) = (3, 2)$, $C_{R_3 \setminus \{v_7\}}(v_5) = (3, 3)$, $C_{R_3 \setminus \{v_7\}}(v_6) = (2, 2)$, and $C_{R_3 \setminus \{v_7\}}(v_7) = (1, 2)$. It is to be noted that, in all three cases, $R_3 \setminus \{v_1\}$, $R_3 \setminus \{v_2\}$, and $R_3 \setminus \{v_7\}$ the other vertices are represented by a unique vector. Therefore, R_3 is a fault-tolerant basis.

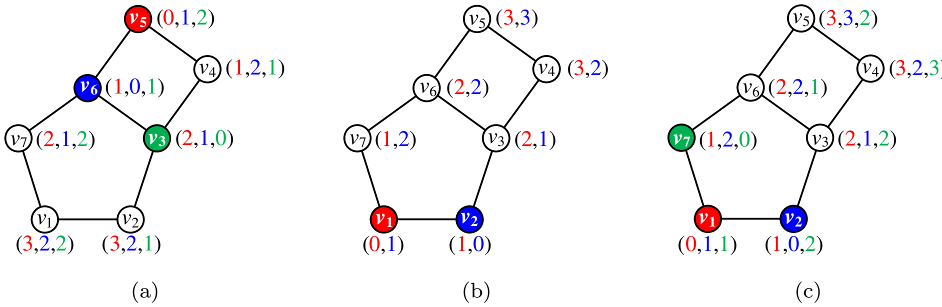


Figure 3: (a) Resolving set; (b) Basis; (c) Fault-tolerant basis

Table 1: Nomenclature

Notations	Description
$V(G)$	Vertex set of G
$E(G)$	Edge set of G
$d(s,t)$	Distance between two vertices s and t
$N_G(s)$	Open neighborhood of s
$N_G[s]$	Closed neighborhood of s
FTMD	Fault-tolerant metric dimension
$\dim(G)$	Metric dimension of G
$\dim'(G)$	Fault-tolerant metric dimension of G
$C_R(s)$	$\|R\|$-vector or the code of s with respect to the set R

The concept of *locating sets* was first introduced by Slater [4,5], who was motivated by how it could be used to insert a few sonar-detecting devices in a network while enabling the position of each vertex to be uniquely identified. He used the terms reference set and location number to describe the minimum resolving set and its cardinality, respectively. This idea was independently found by Harary and Melter, who chose to employ the term metric dimension in lieu of location number [3]. In a subsequent study, Khuller et al. used the word metric dimension to describe these ideas that they independently developed. These ideas were re-investigated by Chartrand et al. [10] and Johnson [11], who were working to build a tool that could handle huge quantities of chemical graphs.

In [2], the problem of finding metric dimension is classified as NP-hard problem for the general graph. It remains NP-complete when restricted to bipartite graphs [12]. It also remains NP-complete even when restricted to planar graphs (including those of bounded degree) [13], split, cobipartite, and line graphs of bipartite graphs [14], directed graphs [15] and permutation graphs or interval graphs [16].

Resolving sets has various uses in tackling problems such as geometrical routing protocols [17], robot navigation, pattern recognition, and image processing [2], network discovery and verification [1], coin weighing problems [18] and connected joins in graphs [19]. Generalized Petersen graphs [20], trees [2], Benes networks [12], enhanced hypercubes [21], honeycomb networks [22], and Illiac networks [23], have all been researched for this problem. Characterization of some graphs with metric dimension two was discussed in [24]. Very recently, k-metric dimension of graphs [25], resolving power domination number of PNN [26], dominant local metric dimension of corona product graphs [27], local multiset dimension of amalgamation graphs [28], local metric dimension of specific types of circulant networks [29], irregular convex triangular networks [30] have been determined.

Hernando et al. developed the idea of FTMD [31]. Javaid et al. rediscovered the same concept of fault-tolerant resolvability in [7]. This problem is investigated for grids [32], circulant graphs [33], square of grids [34], hollow coronoid structure [35], Möbius ladder [36] and convex polytopes [37]. Recently, Prabhu et al. in [38] reinvestigated this fault-tolerant resolvability for multistage interconnection networks that were wrongly computed in [39]. In [40], the fractal cubic network is redefined, and the metric and fault-tolerant metric dimensions are determined.

3 Parallel Interconnection Networks

Interconnection networks play a vital role in facilitating effective communication across processors in a parallel computing system [41]. Two primary methods for connecting these processors are a static high-speed interconnection network and a dynamic interconnection network. High-speed refers to a singular CPU, memory module, or a collection of processors. The objective of high-speed computer networks is to provide rapid and effective communication between nodes. The infrastructure facilitates the transmission of large amounts of data with little delay. Static connectivity networks are immovable. A unidirectional static interconnection network is a type of network where the connections between nodes only enable communication in one way. Data can be sent from one node to another, but not in the opposite direction. In a bidirectional static interconnection network, nodes are connected in a way that enables communication to occur in both ways. The selection between either connection is contingent upon the unique demands of the parallel computing system. In contrast to a static interconnection network (fixed interconnection network), which has permanent connections between nodes, a dynamic interconnection network allows for flexible reconfiguration to accommodate changing communication needs. In this paper, we discuss two fixed interconnection networks called generalized fat trees and the amalgamation of perfect binary trees.

3.1 Generalized Fat Trees

In order to balance traffic flows among the available links and lower the likelihood of congestion, multipath routing methods can use the fat tree, which provides many shortest-path routes between any pair of end nodes. In high-performance interconnection networks, fat-tree is one of the most widely used topologies, with the goals of low latency, effective group communication, and scalability. From a commercial standpoint, the mesh and hypercube topologies are two of these interconnection networks' most well-liked networks. Despite being an effective network due to its symmetry, regularity, logarithmic diameter, modularity, and high fault tolerance [42], since the node degree of the hypercube is not constant, there are issues with packing and wireability for VLSI implementation. A network with a constant node degree can be found in many scientific and engineering issues, including matrix problems, image processing algorithms, and multi-grid methods [43]. Mesh networks are of this kind but have the disadvantage of higher diameter and a lower edge bisection [44]. A good interconnection network needs to have a low node degree, according to [45]. Consequently, a new class of network called generalized fat trees, developed in [46], encompasses various specific instances, such as the fat trees utilized in the pruned butterflies, the CM-5 connection machine architecture, and various other fat trees proposed in the existing literature. This architecture gives designers and analysts of fat tree-based architectures a formal, overarching notion to work with. Leiserson suggested that these networks are hardware efficient [47]. The KSR-1 parallel machine was developed by Kendall Square Research [48]. In [49], an alternative fat tree topology known as the pruned butterfly is proposed, while [50] provides an informal description of more variations. In these topologies, the expansion of channel bandwidth is altered from the initial fat trees in [47]. Now, let us formally recall the definition of the generalized p-ary fat tree.

For a generalized p-ary fat tree $GFT(l,p,q)$, the vertex set $V(GFT(l,p,q)) = \{x^k_{h_m} : 0 \leq h \leq l,\ 1 \leq k \leq p^{l-h},\ 1 \leq m \leq q^h\}$, where m is the location of the vertex in the sub-fat tree of level h, k is the location of copies of level h sub-fat trees, and h is the specific level.

A generalized p-ary fat tree $GFT(l+1,p,q)$ is recursively constructed from p copies of $GFT(l,p,q)$, as $GFT^k(l,p,q) = G(V^k_l, E^k_l)$, $1 \leq k \leq p$, and q^{l+1} extra nodes are introduced, with the condition that each upper-level node $x^k_{l_m}$, for $1 \leq m \leq q^l$, is connected to a sequence of q newly added upper-level nodes consecutively, given by $x^1_{l_{(m-1)q+1}}, x^1_{l_{(m-1)q+2}}, \ldots, x^1_{l_{(m-1)q+(q-1)}}, x^1_{l_{mq}}$. The vertex set $V(GFT(0,p,q)) = \{x^1_{0_1}\}$ and $E(GFT(0,p,q)) = \emptyset$. The graph $GFT^k(l,p,q)$ is a sub-fat tree of $GFT(l+1,p,q)$. See Figures 4 and 5.

In the fat tree architecture, the leaf nodes contain the processing components, and the intermediate nodes act as switches or routers. As a result, $GFT(l,p,q)$ comprises p^l processors at the leaf level and routers or switching nodes at non-leaf levels. Leaf nodes are considered to be in level 0. Every non-root has q parent nodes, while every non-leaf has p children. The level of a vertex s is h if we have a vertex t of level 0 with $d(s,t) = h$. It's evident that a $GFT(l,p,q)$ structure incorporates p^{l-h} sub-fat trees $GFT(h,p,q)$, where $1 \leq h \leq l$. These sub-fat trees are labeled as $GFT^j(h,p,q)$, with $1 \leq j \leq p^{l-h}$, and they collectively contain $p^{l-h} \cdot q^h$ vertices positioned at level h.

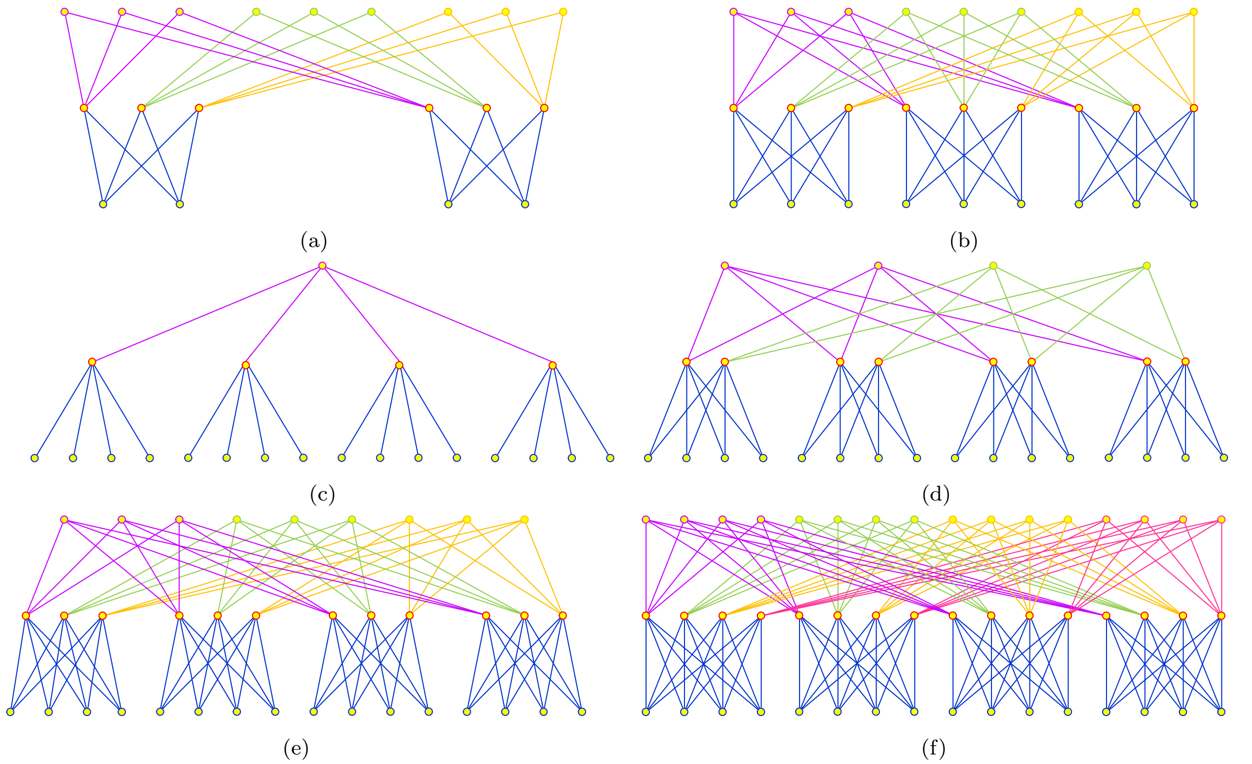


Figure 4: (a) $GFT(2,2,3)$; (b) $GFT(2,3,3)$; (c) $GFT(2,4,1)$; (d) $GFT(2,4,2)$; (e) $GFT(2,4,3)$; (f) $GFT(2,4,4)$

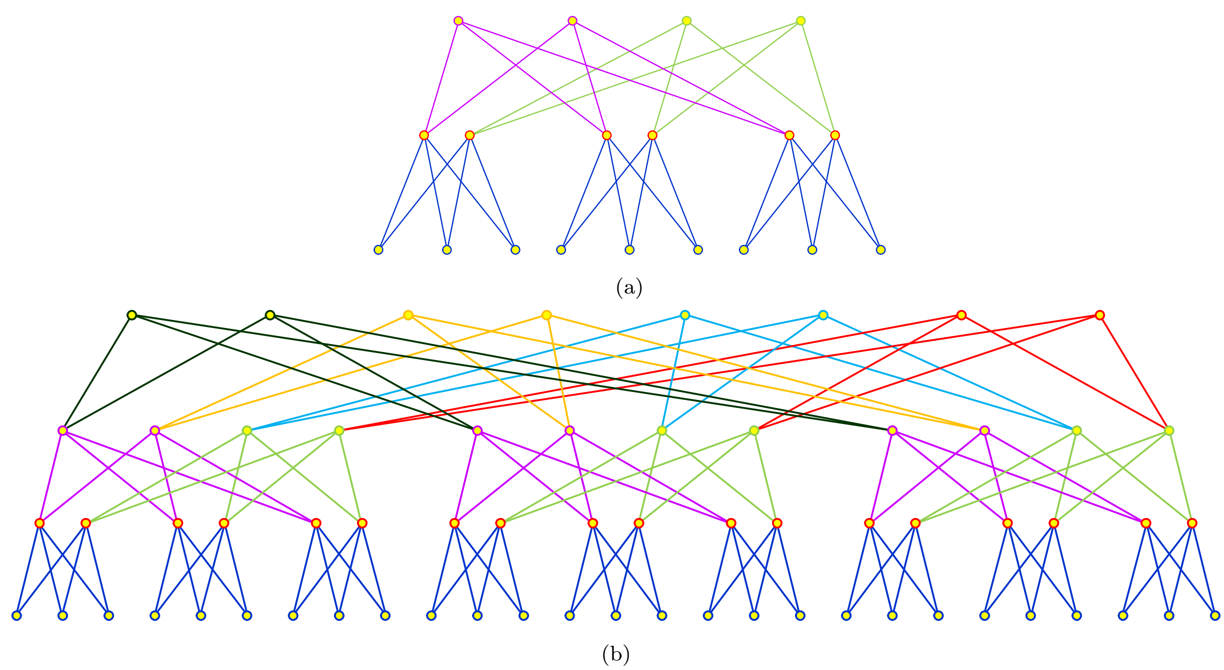


Figure 5: (a) $GFT(2,3,2)$; (b) $GFT(3,3,2)$

3.2 Amalgamation of Perfect Binary Trees

Exploring perfect binary trees provides valuable insights into fundamental principles of data structures, algorithmic design, and the analysis of computational complexity. In databases or search structures, combining index trees or search trees through amalgamation can lead to faster query processing times by reducing the number of tree traversal operations required. In parallel or distributed computing environments, merging trees can facilitate the efficient aggregation of partial computation results from multiple processing units or distributed systems. The purpose of tree amalgamation is to streamline data representation, improve algorithmic efficiency, and facilitate more effective processing of tree-like structures in various computational tasks. Depending on the specific algorithms or operations being performed on the perfect binary trees, amalgamating the leaf nodes could simplify or optimize certain computations. For example, if the trees are being used in search or traversal algorithms, combining leaf nodes could reduce the number of comparisons or operations needed. Now, let us formally recall the definition of the binary tree and perfect binary tree.

A binary tree is described as a tree with only one vertex of degree two, the root vertex, with all other vertices being either one or three in degree. Any binary tree that has all of its leaves at the same level or within a uniform distance from the root vertex is said to be a perfect binary tree. This tree has $2^{l+1} - 1$ vertices where l is the height of the tree.

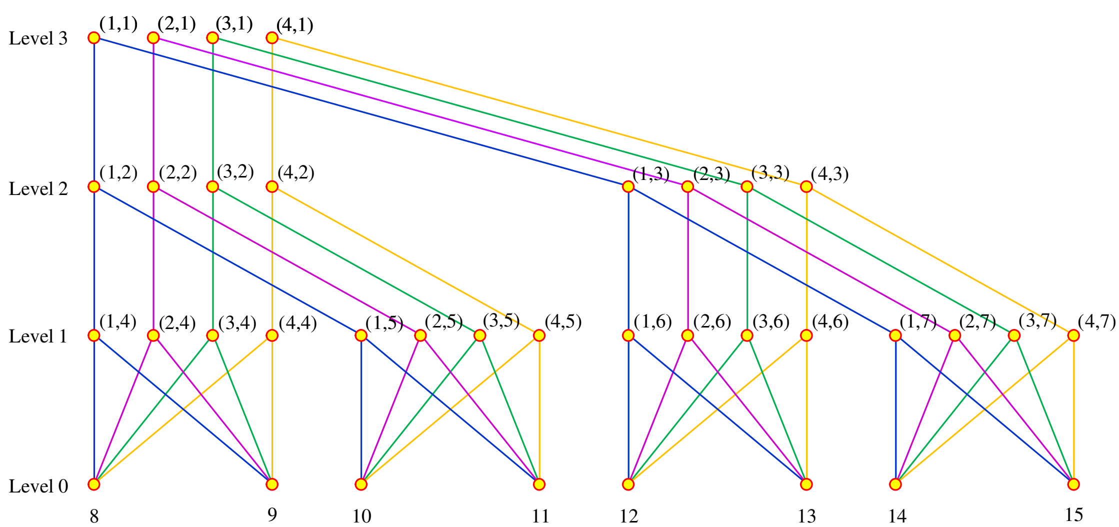


Figure 6: $AT(3, 4)$

In this subsection, we introduce a new architecture called the *amalgamation of perfect binary tree* and list a few of its topological properties. An amalgamation tree $AT(l, w)$ is obtained from w copies of perfect binary trees of height l by identifying the corresponding leaf vertices. We shall call these leaf vertices as *vertices of amalgamation*. Each copy is denoted by T_i^l, $1 \leq i \leq w$. $AT(l, 1)$ is just a perfect binary tree of height l. $AT(l, 2)$ is called a *diamond tree* in the literature. As in the generalized fat trees, leaf vertices are at level 0. A vertex of level h is at a distance h from a descendent leaf vertex. We propose a labeling of vertices of $AT(l,w)$ using the labeling of the vertices of the perfect binary tree. Let $V(AT(l,w)) = \{(i,j) : 1 \leq i \leq w, 2^{l-h} \leq j \leq 2^{l-h+1}-1, 1 \leq h \leq l\} \cup \{2^l, 2^l+1, \ldots, 2^l+(2^l-1)\}$. In the ordered pair (i,j), the first component i refers to the i-th perfect binary tree T_i^l and the second component j denotes the label of a vertex in T_i^l. For every $1 \leq i \leq w$, we fix $B_i = \{(i,t) : 1 \leq t \leq 2^l - 1\}$. See Figure 6.

4 Main Results

The following lemmas were proved in different articles. Though these lemmas explain the necessary vertices to form a basis and fault-tolerant basis, sometimes these vertices may not be sufficient to form a basis and fault-tolerant basis. For example, the authors can see the basis of the probabilistic neural network (**PNN**) discussed in [26]. In **PNN**, the necessary vertices are not sufficient to form a basis.

Lemma 1. [51] *Let* $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ *be the twin sets of* G*, then* $\dim(G) \geq \sum_{i=1}^{n} |\mathcal{T}_i| - n$.

Lemma 2. [38] *If* F *is a fault-tolerant resolving set of* G*, and* t *has a twin in* G*, then* $t \in F$*. Moreover, if* S *is the union of all twin sets in* G*, then it follows that* $\dim'(G) \geq |S|$.

Theorem 3. [52] $\dim(GFT(l,p,q)) = (p-1)p^{l-1} + (q-1)q^{l-1}$ *where* $p, q \geq 2$ *and* $l \geq 2$.

Now, we discuss the fault-tolerant resolving number of generalized fat trees. For convenience, let us denote the vertices of level 0 by T_A and that of level l by T_B.

Remark. *The sets* T_A *and* T_B *can be partitioned into* T_{A_i} *and* T_{B_j} *where*
$T_{A_i} = \{x_{0_1}^{(i-1)p+1}, x_{0_1}^{(i-1)p+2}, \ldots, x_{0_1}^{(i-1)p+p}\}$, $1 \leq i \leq p^{l-1}$ *and* $T_{B_j} = \{x^1_{l+1_{(j-1)q+1}}, x^1_{l+1_{(j-1)q+2}}, \ldots, x^1_{l+1_{(j-1)q+q}}\}$, $1 \leq j \leq q^{l-1}$. *Also,* $|T_{A_i}| = p$ *and* $|T_{B_i}| = q$.

Theorem 4. $\dim'(GFT(l,p,q)) = p^l + q^l$ *where* $p, q \geq 2$ *and* $l \geq 2$.

Proof. $GFT(l,p,q)$ contains p copies of $GFT(l-1,p,q)$ represented by $GFT^k(l-1,p,q)$, $1 \leq k \leq p$. The number of vertices in level 0 is p^l, and the number of vertices in level l is q^l. It has $p^{l-h} \cdot q^h$ vertices at any level h in general, and these are the vertices in the top level of p^{l-h} copies of $GFT(h,p,q)$. For every pair $u, v \in T_{A_i}$, $N(u) = N(v)$, which implies u and v are twins. Similarly for $u, v \in T_{B_j}$, $N(u) = N(v)$. Therefore, each T_{A_i} and T_{B_j} are twin sets. See Figure 7. By Lemma 2, $\dim'(GFT(l,p,q)) \geq |\cup_{i=1}^{p^{l-1}} T_{A_i} \bigcup \cup_{i=1}^{q^{l-1}} T_{B_j}| = p^l + q^l$.

We next claim that $\dim'(GFT(l,p,q)) \leq p^l + q^l$. Let $F \coloneqq T_A \cup T_B$ and $x, y \in V(GFT(l,p,q)) \setminus F$.

Case 1: x and y are at level h, $0 < h < l$.

Case 1.1: $x = u_{h_{m_1}}^{k_1}$ and $y = u_{h_{m_2}}^{k_2}$, $1 \leq m_1, m_2 \leq q^h$, $1 \leq k_1, k_2 \leq p^{l-h}$, and $k_1 \neq k_2$.
Corresponding to each vertex $x = u_{h_m}^k \in V(GFT(l,p,q))$, we define a set $A_x = \bigcup_{j=1}^{p^{h-1}} T_{A_{(k-1)p^{h-1}+j}}$. It is obvious that every A_x is a subset of F and $\forall s \in A_x$, $d(x,s) = h$ and $d(y,s) > h$. Similarly, $\forall t \in A_y$, $d(y,t) = h$ and $d(x,t) > h$.

Case 1.2: $x = u^{k_1}_{h_{m_1}}$ and $y = u^{k_2}_{h_{m_2}}$, $1 \le m_1, m_2 \le q^h$, $1 \le k_1, k_2 \le p^{l-h}$, $k_1 = k_2$ and $m_1 \ne m_2$.

Corresponding to each vertex $x = u^k_{h_m} \in V(GFT(l,p,q))$ where $h = l-1-i, 0 \le i \le l-2$, we define a set $B_x = \bigcup_{j=1}^{q^i} T_{B_{(m-1)q^i+j}}$. It is obvious that every B_x is a subset of F and $\forall s \in B_x$, $d(x,s) = l-h$ and $d(y,s) > l-h$. Similarly, $\forall t \in B_y$, $d(y,t) = l-h$ and $d(x,t) > l-h$.

Case 2: x and y are in different levels

Assume that x is in level h_1 and y is in level h_2 and $h_1 < h_2$. There exists $u \in A_x$ such that $d(x,u) = h_1$ and $d(y,u) > h_1$. Similarly there exists $v \in B_y$ such that $d(y,v) = l-h_2$ and $d(x,v) > l-h_2$.

This proves that F is a fault-tolerant resolving set. □

It is worth noting that $GFT(l,p,1)$ and $GFT(l,1,q)$ are isomprphic whenever $p = q$.

Theorem 5. $\dim'(GFT(l,p,1)) = p^l$.

Proof. For every i, T_{A_i} is a twin set. Therefore, by Lemma 2, $\dim'(GFT(l,p,1)) \ge p^l$. We next claim that $\dim'(GFT(l,p,1)) \le p^l$. To do this, we demonstrate a fault-tolerant metric basis set of cardinality p^l. Consider two vertices $x, y \in V(GFT(l,p,1)) \setminus F$.

Case 1: x and y are level h vertices, $0 < h < l$.

In this case, $x = u^{k_1}_{h_m}$ and $y = u^{k_2}_{h_m}$, $m = 1$, $1 \le k_1, k_2 \le p^{l-h}$, and $k_1 \ne k_2$.

We define a set $A_x := \bigcup_{j=1}^{p^{h-1}} T_{A_{(k-1)p^{h-1}+j}}$ for each corresponding vertex $x = u^k_{h_m} \in V(GFT(l,p,1))$. It is obvious that every A_x is a subset of F and $\forall s \in A_x$, $d(x,s) = h$ and $d(y,s) > h$. Similarly, $\forall t \in A_y$, $d(y,t) = h$ and $d(x,t) > h$.

Case 2: x and y are in different levels

Let's assume that x is in level h_1 and y is in level h_2 and $h_1 < h_2$. There exists $u \in A_x$ such that $d(x,u) = h_1$ and $d(y,u) > h_1$. Similarly there exists $v \in A_y$ such that $d(y,v) = h_2$ and $d(x,v) \ne h_2$.

This proves that F is a fault-tolerant resolving set and $\dim'(G(l,p,1)) \le p^l$. □

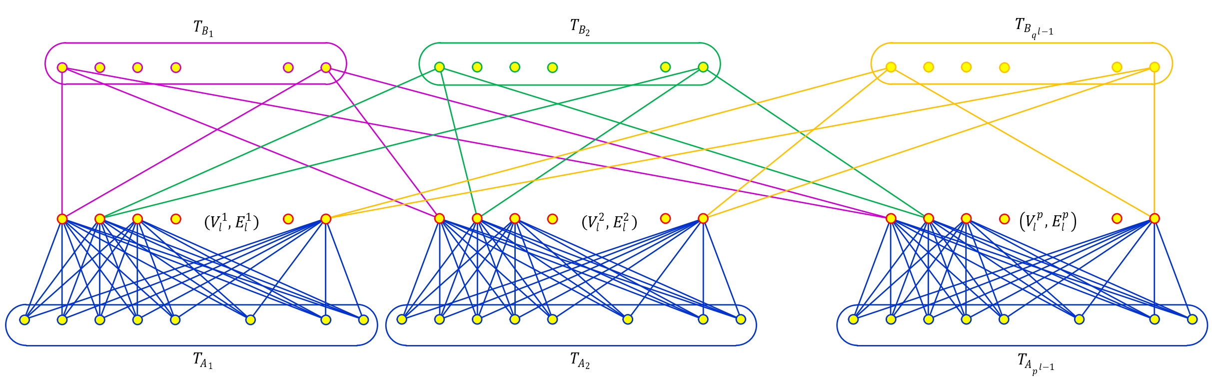


Figure 7: Twin sets in $GFT(l,p,q)$

Theorem 6. [52] *Let G be $AT(l,w)$, $w \ge 2$. Then* $\dim(G) \le 2^{l-1} + w - 1$.

Since $AT(l,1)$ is isomorphic to a perfect binary tree of height l and for which the result is readily available in [2], we state the following results for $w \geq 2$.

Lemma 7. *Let G be $AT(l,w), w \geq 2$ and R be a resolving set of G, then $R \cap (B_i \cup B_j) \neq \emptyset$ whenever $i \neq j$ and $1 \leq i, j \leq w$.*

Proof. Suppose $R \cap (B_i \cup B_j) = \emptyset$, then we have $d(x,(i,t)) = d(x,(j,t))$ for each $x \in R$ and $1 \leq t \leq 2^l - 1$. This contradicts our assumption that R is a resolving set. □

Theorem 8. *Let G be $AT(l,w)$, $w \geq 2$. Then $\dim(G) = 2^{l-1} + w - 1$.*

Proof. The vertices of amalgamation form 2^{l-1} twin sets with two vertices in each twin set. Lemma 1 implies that the 2^{l-1} vertices, one vertex from each twin set, are necessary to form a resolving set for G. Lemma 7 guarantees that, along with the previously mentioned 2^{l-1} vertices, an additional $w-1$ vertices are required.

By utilizing Theorem 6 and the above arguments, we can conclude that $\dim(G) = 2^{l-1} + w - 1$. □

Continuing the computation of the metric dimension of amalgamation of perfect binary trees, we now investigate the fault-tolerant metric dimension for this architecture.

Theorem 9. *Let G be $AT(l,w)$, $w \geq 2$. Then $\dim'(G) = 2^l + w$.*

Proof. As we have witnessed already, the vertices of amalgamation are twin vertices, and it follows from Lemma 2 that any fault-tolerant metric basis of $AT(l,w)$ must contain 2^l vertices of level 0. Lemma 7 aids in claiming that an additional w vertices are also required, which leads to affirming that $\dim'(G) \geq 2^l + w$.

We present a fault-tolerant resolving set of cardinality $2^l + w$. Let F_1 be the set of all vertices of level 0 and $F_2 = \{(i,1), 1 \leq i \leq w\}$. We claim that $F = F_1 \cup F_2$ is a fault-tolerant resolving set for G. Let $x, y \in V \setminus F$. We consider different cases depending upon whether x and y belong to the same copy of the binary tree or different copies and whether they are in the same level or different levels. Let $x = (i_1, j_1)$, $y = (i_2, j_2)$ where $1 \leq i_1, i_2 \leq w$, $2^{l-h} \leq j_1, j_2 \leq 2^{l-h+1} - 1$, $1 \leq h \leq l$.

Case 1: $i_1 = i_2$

Let (i_1, j_1) and (i_2, j_2) be the vertices of level h_1 and h_2, respectively.

Case 1.1: $h_1 = h_2$

There exists $k_1, k_2 \in F_1$ such that $d((i_1,j_1),k_1) = h_1$ and $d((i_2,j_2),k_1) > h_1$.
$d((i_1,j_1),k_2) = h_1$ and $d((i_2,j_2),k_2) > h_1$.

Case 1.2: $h_1 < h_2$

There exists $k_1, k_2 \in F_1$ such that $d((i_1,j_1),k_1) = h_1$ and $d((i_2,j_2),k_1) > h_1$.
$d((i_1,j_1),k_2) = h_1$ and $d((i_2,j_2),k_2) > h_1$.

Case 2: $i_1 \neq i_2$

For every (i_1,j_1) and (i_2,j_2), there exists $(i_1,1),(i_2,1) \in F_2$ such that $d((i_1,1),(i_1,j_1)) < d((i_1,1),(i_2,j_2))$ and $d((i_2,1),(i_1,j_1)) < d((i_2,1),(i_2,j_2))$.

This proves our claim that F is a fault-tolerant resolving set and thus $\dim'(G) \leq 2^l + w$.

$\square$

The above theorem shows that the fault-tolerant metric dimension increases by one for every perfect binary tree amalgamation. Table 2, shows the numerical values of $\dim(AT(l,w))$ and $\dim'(AT(l,w))$. It is evident that the amalgamation of each perfect binary tree increases the order of the graph by $2^l - 1$, but both the parameters $\dim(AT(l,w))$ and $\dim'(AT(l,w))$ are increased by just one.

Table 2: Numerical values of $\dim(AT(l,w))$ and $\dim'(AT(l,w))$

l	$\lvert V(AT(l,w))\rvert$	$\dim(AT(l,w))$	$\dim'(AT(l,w))$
3	$8(w+1)-w$	$w+3$	$w+8$
4	$16(w+1)-w$	$w+7$	$w+16$
5	$32(w+1)-w$	$w+15$	$w+32$

5 Conclusion and Future Direction

The metric dimension is an intuitively straightforward concept. The application of trilateration in continuous space and its close relationship with GPS make it suitable for locating graph nodes. The metric dimension is important in telecommunication networks, including cable networking, fibre optics, and CCTV cameras. The metric dimension can assist us in reducing time, labour, and cost in the aforementioned networks and increasing their effectiveness. Additional research in this area could eventually result in a better understanding of the metric dimension and its application to real-world networks. The fat-tree interconnection network is one of the topologies utilised in supercomputers because of its high bisection bandwidth and simplicity of application mapping for multiple communication topologies. Most applications only need a fraction of the connectivity that fat trees can give regarding communication topology. Fat trees are particularly well-suited for using methods for lowering power usage since they offer several alternative paths for each source/destination combination. In this article, we solve the FTMD of generalized fat trees. We have found that $\dim'(GFT(l,p,q)) - \dim(GFT(l,p,q)) = p^{l-1} + q^{l-1}$. We also state that the metric dimension and FTMD for amalgamation of perfect binary trees with $w \geq 2$ differ by $2^{l-1} + 1$.

The motivation behind the problem of finding a metric basis for a network is to identify its vertices uniquely. Thorough monitoring of each vertex is now in progress with the help of a metric basis. However, if an intruder gains access to the network by exploiting connections between vertices (edges) rather than directly through them, it becomes impossible to identify such an intruder. In this case, the surveillance system fails to fulfil its purpose, and further measures are needed to secure the network. This opened up the concept of a novel variation of metric basis known as edge metric basis. Though we have characterizations and numerous techniques to find the metric dimension and fault-tolerant metric dimension, there is no literature about characterization or efficient lower bounds on the edge metric dimension and fault-tolerant edge metric dimension of a graph. It will be a challenging open problem for future researchers to investigate edge metric dimension and fault-tolerant edge metric dimension characterizations and lower bounds.

Compliance with ethical standards

Conflict of interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Data Availability Statement: No Data associated with the manuscript.

References

[1] Z. Beerliova, F. Eberhard, T. Erlebach, A. Hall, M. Hoffman, M. Mihalák, Network discovery and verification, IEEE Journal on Selected Areas in Communications **24**(12) (2006) 2168–2181.

[2] S. Khuller, B. Ragavachari, A. Rosenfield, Landmarks in graphs, Discrete Applied Mathematics **70**(3) (1996) 217–229.

[3] F. Harary, R.A. Melter, On the metric dimension of a graph, Ars Combinatoria **2** (1976) 191–195.

[4] P.J. Slater, Leaves of trees, Congressus Numerantium **14** (1975) 549–559.

[5] P.J. Slater, Dominating and reference sets in a graph, Journal of Mathematical and Physical Sciences **22**(4) (1988) 445–455.

[6] G. Chartrand, P. Zhang, The theory and applications of resolvability in graphs, a survey, Congressus Numerantium **160** (2003) 47–68.

[7] I. Javaid, M. Salman, M.A. Chaudhry, S. Shokat, Fault-tolerance in resolvability, Utilitas Mathematica **80** (2009) 263–275.

[8] M. Salman, I. Javaid, M.A. Chaudhry, Minimum fault-tolerant, local and strong metric dimension of graphs, Ars Combinatoria **138** (2018) 333–353.

[9] W.H. Wang, M. Palaniswami, S.H. Low, Optimal flow control and routing in multi-path networks, Performance Evaluation **52** (2003) 119–132.

[10] G. Chartrand, L. Eroh, M.A. Johnson, O. Oellermann, Resolvability in graphs and the metric dimension of a graph, Discrete Applied Mathematics **105** (2000) 99–113.

[11] M. Johnson, Structure-activity maps for visualizing the graph variables arising in drug design, Journal of Biopharmaceutical Statistics **3**(2) (1993) 203–236.

[12] P. Manuel, M.I. Abd-El-Barr, I. Rajasingh, B. Rajan, An efficient representation of Benes networks and its applications, Journal of Discrete Algorithms **6**(1) (2008) 11–19.

[13] J. Díaz, O. Pottonen, M. Serna, and E. J. Van Leeuwen, On the complexity of metric dimension, in European Symposium on Algorithms, Springer, 2012, pp. 419–430.

[14] L. Epstein, A. Levin, and G. J. Woeginger, The (weighted) metric dimension of graphs: Hard and easy cases, Algorithmica **72** (2015) 1130–1171.

[15] B. Rajan, I. Rajasingh, J.A. Cynthia, and P. Manuel, Metric dimension of directed graphs, International Journal of Computer Mathematics **91**(7) (2014), 1397–1406.

[16] F. Foucaud, G. B. Mertzios, R. Naserasr, A. Parreau, and P. Valicov, Identification, location-domination and metric dimension on interval and permutation graphs. II. Algorithms and complexity, Algorithmica **78** (2017) 914–944.

[17] K. Liu, N. Abu-Ghazaleh, Virtual coordinates with backtracking for void traversal in geographic routing, Lecture Notes in Computer Science **4104** (2006) 46–59.

[18] S. Söderberg, H.S. Shapiro, A combinatory detection problem, The American Mathematical Monthly **70**(10) (1963) 1066–1070.

[19] A. Sebő, E. Tannier, On metric generators of graphs, Mathematics of Operations Research **29**(2) (2004) 383–393.

[20] S. Ahmad, M.A. Chaudhry, I. Javaid, M. Salman, On the metric dimension of generalized Petersen graphs, Quaestiones Mathematicae **36**(3) (2013) 421–435.

[21] B. Rajan, I. Rajasingh, M.C. Monica, P. Manuel, Metric dimension of enhanced hypercube networks, Journal of Combinatorial Mathematics and Combinatorial Computing **67** (2008) 5–15.

[22] P. Manuel, B. Rajan, I. Rajasingh, M.C. Monica, On minimum metric dimension of honeycomb networks, Journal of Discrete Algorithms **6**(1) (2008) 20–27.

[23] B. Rajan, I. Rajasingh, P.V. Gopal, M.C. Monica, Minimum metric dimension of Illiac networks, Ars Combinatoria **117** (2014) 95–103.

[24] A. Behtoei, A. Davoodi, M. Jannesari, and B. Omoomi, A characterization of some graphs with metric dimension two, Discrete Mathematics, Algorithms and Applications **9**(2) (2017) 1750027.

[25] I.G. Yero, A. Estrada-Moreno, J.A. Rodríguez-Velázquez, Computing the k-metric dimension of graphs, Applied Mathematics and Computation **300** (2017) 60–69.

[26] S. Prabhu, S. Deepa, M. Arulperumjothi, L. Susilowati, J.B. Liu, Resolving-power domination number of probabilistic neural networks, Journal of Intelligent & Fuzzy Systems **43**(5) (2022) 6253–6263.

[27] R. Umilasari, L. Susilowati, Slamin, S. Prabhu, On the dominant local metric dimension of corona product graphs, IAENG International Journal of Applied Mathematics **52**(4) (2022) IJAM-52-4-38.

[28] R. Alfarisi, L. Susilowati, D. Dafik, S. Prabhu, Local multiset dimension of amalgamation graphs, F1000Research **12** (2023) 95.

[29] V.J.A. Cynthia, M. Ramya, S. Prabhu, Local metric dimension of certain classes of circulant networks, Journal of Advanced Computational Intelligence and Intelligent Informatics **27**(4) (2023) 554–560.

[30] S. Prabhu, D. Sagaya Rani Jeba, M. Arulperumjothi and S. Klavžar, Metric dimension of irregular convex triangular networks, AKCE International Journal of Graphs and Combinatorics (2023), DOI: 10.1080/09728600.2023.2280799.

[31] C. Hernando, M. Mora, P.J. Slater, D.R. Wood, Fault-tolerant metric dimension of graphs, Convexity in Discrete Structures in Ramanujan Mathematical Society Lecture Notes **5** (2008) 81–85.

[32] A. Simić, M. Bogdanović, Z. Maksimović, J. Milošević, Fault-tolerant metric dimension problem: A new integer linear programming formulation and exact formula for grid graphs, Kragujevac Journal of Mathematics **42**(4) (2018) 495–503.

[33] M. Basak, L. Saha, G.K. Das, K. Tiwary, Fault-tolerant metric dimension of circulant graphs $C_n(1,2,3)$, Theoretical Computer Science **817** (2020) 66–79.

[34] L. Saha, M. Basak, K. Tiwary, All metric bases and fault-tolerant metric dimension for square of grid, Opuscula Mathematica **42**(1) (2022) 93–111.

[35] A.N.A. Koam, A. Ahmad, M.E. Abdelhag, M. Azeem, Metric and fault-tolerant metric dimension of hollow coronoid, IEEE Access **2021** (2021) 81527–81534.

[36] H. Wang, M. Azeem, M.F. Nadeem, A. Ur-Rehman, A. Aslam, On fault-tolerant resolving sets of some families of ladder networks, Complexity **2021** (2021) 9939559.

[37] H. Raza, S. Hayat, X.F. Pan, On the fault-tolerant metric dimension of convex polytopes, Applied Mathematics and Computation **339** (2018) 172–185.

[38] S. Prabhu, V. Manimozhi, M. Arulperumjothi, S. Klavžar, Twin vertices in fault-tolerant metric sets and fault-tolerant metric dimension of multistage interconnection networks, Applied Mathematics and Computation **420** (2022) 126897.

[39] S. Hayat, A. Khan, M.Y.H. Malik, M. Imran, M.K. Siddiqui, Fault-tolerant metric dimension of interconnection networks, IEEE Access **8** (2020) 145435–145445.

[40] M. Arulperumjothi, S. Klavžar, S. Prabhu, Redefining fractal cubic networks and determining their metric dimension and fault-tolerant metric dimension, Applied Mathematics and Computation **452** (2023) 128037.

[41] D. Lüdtke and D. Tutsch, The modeling power of CINSim: Performance evaluation of interconnection networks, Computer Networks **53** (2009) 1274–1288.

[42] Y. Saad, M.H. Schultz, Topological properties of hypercubes, IEEE Transactions on Computers **37**(7) (1988) 867–872.

[43] T.F. Chan, Y. Saad, Multigrid algorithms on the hypercube multiprocessor, IEEE Transactions on Computers **35**(11) (1986) 969–977.

[44] K. Hwang, J. Ghosh, Hypernet: A communication-efficient architecture for constructing massively parallel computers, IEEE Transactions on Computers **36**(12) (1987) 1450–1466.

[45] J. Xu, Topological structures and analysis of interconnection networks, Springer Science & Business Media **7** (2013).

[46] S.R. Ohring, M. Ibel, S.K. Das, M.J. Kumar, On generalized fat trees, Proceedings of 9th International Parallel Processing Symposium (1995) pp. 37–44.

[47] C.E. Leiserson, Fat-trees: Universal networks for hardware-efficient supercomputing, IEEE Transactions on Computers **34**(10) (1985) 892–901.

[48] S. Frank, H. Burkhardt, J. Rothnie, The KSR 1: bridging the gap between shared memory and MPPs, Digest of Papers. Compcon Spring, (1993) pp. 285–294.

[49] P. Bay, G. Bilardi, Deterministic on-line routing on area-universal networks, Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science (1990) pp. 297–306.

[50] R.I. Greenberg, C.E. Leiserson, Randomized routing on fat trees, JAI Press **5** (1989) 345–374.

[51] S. Prabhu, T. Flora, M. Arulperumjothi, On independent resolving number of $TiO_2[m, n]$ nanotubes, Journal of Intelligent & Fuzzy Systems **35**(6) (2018) 6421–6425.

[52] B. Rajan, A. William, S. Prabhu, On certain resolving parameters of tree derived architectures, Journal of Combinatorial Mathematics and Combinatorial Computing **92** (2015) 233–242.