Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

Eccentricity Centrality of the Comb Product Between Well-Known Graphs and Interval Graphs: Applications in Warehouse Network Optimisation

Shaoli Nandi[1,2], Sukumar Mondal[3], Sovan Samanta[4,5,6], Sambhu Charan Barman[7], Leo Mrsic[5,8], Antonios Kalampakas[9*], Tofigh Allahviranloo[4,10]

[1]Research Centre in Natural and Applied Sciences, Department of Mathematics, Raja N. L. Khan Women's College (Autonomous), Midnapore, 721102, India
[2]Department of Mathematics, Government General Degree College, Salboni, 721516, India
[3]Department of Mathematics, Raja N. L. Khan Women's College (Autonomous), Midnapore, 721102, India
[4]Research Center of Performance and Productivity Analysis, Istinye University, Istanbul, 34320, Turkey
[5]Department of Technical Sciences, Algebra Bernays University, Gradiscanska 24, Zagreb, 10000, Croatia
[6]Department of Technical Sciences, Western Caspian University, Baku, 1001, Azerbaijan
[7]Department of Mathematics, Shahid Matangini Hazra Government General Degree College for Women, Purba Medinipur, 721649, India
[8]Rudolfovo Science and Technology Centre, Podbreznik 15, Navo Mesto, 8000, Slovenia
[9]Department of Mathematics, College of Engineering and Technology, American University of the Middle East, Egaila, 54200, Kuwait
[10]Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran
E-mail: antonios.kalampakas@aum.edu.kw

Received: 9 September 2025; **Revised:** 25 September 2025; **Accepted:** 28 October 2025

Abstract: In network analysis, measuring centrality is essential for determining the relative importance of each vertex within a network. A vertex with higher centrality signifies greater importance compared to others. To facilitate theoretical studies, networks are commonly modelled using graphs. Deoxyribonucleic Acid (DNA) molecules, some scheduling problems, and food webs have a common linear structure that can be modelled as interval graphs. We explore this matter within the framework of calculating vertex eccentricities to ascertain the comparative importance of nodes within the network structure. Eccentricity centrality plays an important role in identifying significant vertices in social networks, facility location networks, etc. In this paper, we compute the eccentricity centrality of the comb product between a well-known graph and an interval graph, and we design two $O(n)$ time algorithms—one for finding the eccentricity of all vertices of the interval graph and another for making a Breadth-First Search (BFS) tree of interval graph. We also compute the eccentricity centrality of the comb product between two interval graphs using these algorithms. We also analyse the time complexity of the proposed algorithms. Finally, we present a real application involving in finding a central warehouse in a warehouse network of an online product-selling company based on our study results.

***Keywords*:** eccentricity centrality, comb product of two graphs, interval graphs

MSC: 05C30, 0562, 05C38

DOI: https://doi.org/10.37256/cm.6620258578

1. Introduction

In network analysis, centrality measurement is a fundamental step to find the correlative importance of every vertex. The higher centrality of a vertex indicates that it is more important than others. For theoretical investigations, it is necessary to represent a network by a graph. In communication networks, such as online social platforms, telecommunications, and extensive brain networks, there is often a delay in message transmission between a sender and a recipient due to the distance between them. The distance $d(u, v)$ represents the shortest path length between nodes u and v in a graph. Measuring the distance between two vertices is particularly relevant in solving facility location problems. For instance, when broadcasting a message, it is ideal to minimize the maximum distance any vertex has from the source. This can be achieved by choosing a vertex with the minimum eccentricity or maximum centrality, optimizing message delivery across the network.

The eccentricity of a vertex u is the greatest distance between u and the other vertices of the graph. The eccentricity centrality of u is defined by the reciprocal of the eccentricity of u. A vertex with lower eccentricity indicates higher centrality of the vertex. Mathematically, the eccentricity centrality $E_C(u)$ is defined by $E_C(u) = \frac{1}{max_{y \in V} d(u, y)}$ where V represents the vertex set in the network. If the eccentricity of a vertex is high, then other vertices of the graph are near. And, if the eccentricity of a vertex is low, then there is at least one vertex that is far from the other. The eccentricity is a more powerful parameter if it is high. The objective of this measure is to identify the vertices that could be reached from other vertices more quickly. Eccentricity centrality is used to analyze many networks [1, 2].

In 1950, Hajos and Benzer first introduced the interval graphs. It plays an essential role in refining the linear arrangement of Deoxyribonucleic Acid (DNA) molecules [3]. They have been applied as mathematical models in various fields, such as food web networks [4] (also known as consumer-resource systems) and job scheduling [5] in industry. The first algorithm capable of recognizing interval graphs in linear time and constructing an interval model was developed using a tree-based data structure [6]. Later, more straightforward linear-time algorithms emerged, utilizing alternative characteristics of interval graphs [7, 8] to improve efficiency.

An Interval Graph (IG) is nothing but an intersection graph of a collection of intervals on a straight line. Let $G = (V, E)$ be a simple, connected, and undirected graph, and $I = \{I_1, I_2, \ldots, I_n\}$ be a set of intervals which are placed on R (the real line). A graph G is referred to as an *interval graph* if there is a function that maps the vertices of G with the intervals in I such that two nodes of G are adjacent if and only if their corresponding intervals overlap. The set I is referred to as the Interval Representation (IR) of the IG G. In this article, we denote an IG G as I_G. Let $I_i = [a_i, b_i]$, $1 \leq i \leq n$, where a_i and b_i represent, respectively, the left endpoint and right endpoint of the interval I_i. We assumed that every interval consists of two endpoints and no two intervals share the same endpoint. We also assumed that the intervals in I are indexed according to their right endpoints in ascending order.

In the area of graph theory, Breadth-First Search (BFS) [9, 10] is a widely used and fundamental algorithm for traversing graphs. This algorithm enables the construction of a BFS tree from a connected graph by visiting all vertices at one level before progressing to the next. In [11], Tarjan introduced an $O(n+m)$ time complexity algorithm to build a BFS tree for general graphs. They [12] also presented an algorithm to construct a BFS tree of IG that takes $O(logn)$ time employing $O(n)$ processors on an Exclusive-Read, Exclusive-Write (EREW) Parallel Random Access Machine (PRAM) model. An IG I_G and its IR are displayed in Figure 1, and a BFS tree rooted at v_1 of I_G is shown in Figure 2. We use these figures throughout the paper. We consider that the interval representation of an interval graph is given.

It is well-known that for a graph consisting of n vertices and m edges, the eccentricities of all vertices can be computed in $O(nm)$ time, by BFS from each vertex. This algorithm is not typically used in real-world scenarios for massive, intricate networks like Facebook, which involve hundreds of millions of nodes and billions of connections. To construct the BFS tree for these cases, it takes hours to complete [13]. In [14], Saha has formulated an efficient algorithm (takes only $O((n^2/p) + logn)$time) for determining the diameter, eccentricity, and radius of circular-arc-graphs under an Exclusive-Read, Exclusive-Write PRAM with p processors. Nandi et al. [15] computed the diameter of the permutation graph by BFS tree within $O(n)$ time. In the present article, we find the eccentricity centrality of the comb product between two

IGs and the comb product between the common graphs (complete graph, wheel graph, star graph, path graph, and cycle graph).

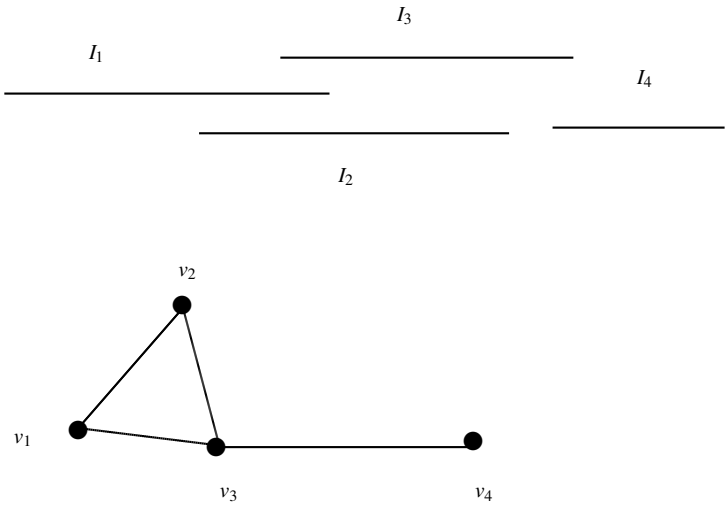

Figure 1. Interval graph I_G and its representation

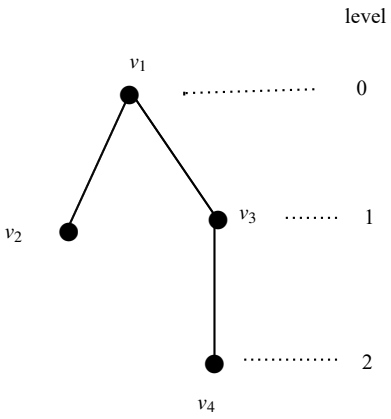

Figure 2. BFS tree of interval graph I_G

1.1 *Survey*

The eccentricity centrality is based on the shortest path between a vertex and other vertices in the networks. Hage and his co-author [16] first explored the idea of eccentricity centrality based on eccentricity in the network. They used the Floyd-Warshall algorithm to obtain the eccentricity, and this algorithm takes $O(n^3)$ time to compute the eccentricity centrality. Authors [17] computed eccentricity of interval graph using the intersection mode, but required $O(n^2)$ time. Olariu [18] designed an $O(m+n)$-time algorithm to find the center (indirectly eccentricity as computation of center depends on it) of an interval graph. Authors [19] developed an algorithm to compute the eccentricity distribution of large graphs. Few authors [20] studied the total eccentricity of some graph operations, and a bound for that of the tensor product. In 2017, Puthuessery et al. [21] computed the eccentricity centrality for connected a connected. Bentert et al. proposed a faster algorithm to determine the centralities of interval + kv graphs [22] constructs on the existence of a quadratic-time algorithm for finding all pair shortest distances on IG [23]. Authors [24] designed efficient shared-memory parallel algorithms (which are applicable in many real-life applications arising in large-scale network analysis) and presented the

first comprehensive experimental study of graph eccentricity computation algorithms in the literature. Also, Li et al. [25] studied the efficiency issue of computing and maintaining the eccentricity distribution on a dynamic small-world network. Authors [26] explored the eccentricities of the vertices of a graph via parallel set cover. In 2022, Ducoffe [27] presented an almost linear time algorithm to find the eccentricities of all vertices (so, all graph centralities) of interval + kv graphs for each fixed k, and he also computed all eccentricities of IG in $O(m+n\log^3 n)$ time. In the same year, Li et al. [28] designed algorithms for the scalable computation of eccentricities of all nodes of graphs. After that, Gomez et al. [29] explored the path eccentricity of graphs. In 2024, Lu et al. [30] studied the resistance eccentricity (that is, useful for measuring the structural significance of a node in network science) of nodes in a graph.

1.2 *Result*

In the present article, we propose a faster sequential algorithm to make a BFS tree of IG within $O(n)$ time. After that, we study some new properties on the BFS tee of IG, the eccentricity of each node of IG and the relation between the height of its BHS tree, and diameter. Based on these results, we compute the eccentricities of all vertices of IG, designing an $O(n)$ time algorithm. Besides these, we find the eccentricity centrality of the comb product between two IGs and the comb product between the common graph (complete graph, wheel graph, star graph, path graph, and cycle graph) and IG. We also analyze the time complexity of the proposed algorithms.

1.3 *Structure of the article*

The next section presents some theorems related to the eccentricity centrality of the comb product between two graphs. In Subsection 2.1, we find the eccentricity centrality of the comb product between a complete graph and an interval graph. In Subsection 2.2, we find the eccentricity centrality of $S_n \rhd I_G$. Subsection 2.3 gives the eccentricity centrality of $W_n \rhd I_G$. The eccentricity centrality of $C_n \rhd I_G$ is calculated in Subsection 2.4. Subsection 2.5 gives the eccentricity centrality of $P_n \rhd I_G$. In Subsection 2.6, we design a sequential algorithm to make a BFS tree of IG. After that, we study some new properties on the BFS tree of IG, the eccentricity of each node of IG, and the relation between the height of its BFS tree and diameter. Here, we also develop an efficient algorithm to compute the eccentricities of all vertices of IG. Finally, we propose a result to find the eccentricity centrality of the comb product between two IGs. The last section provides the paper's conclusion.

2. Eccentricity centrality of comb product of graphs

For two connected graphs G, H, where o is a node of H, the comb product, symbolled by $G \rhd H$, of G and H is a graph made by taking one copy of G and $|V(G)|$ copies of H and grafting the i^{th} copy of H at the node o to the i^{th} node of G.

2.1 *Eccentricity centrality of $K_n \rhd I_G$*

$K_n \rhd I_G$ is the comb product between the complete graph K_n and the interval graph I_G. Suppose that the grafting occurs at v_1 of each copy of I_G. Let the vertices of K_n and I_G are $\{u_i : i = 1 : 1 : n\}$ and $\{v_j : j = 1 : 1 : m\}$ respectively. We re-label the nodes of $K_n \rhd I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}$, $v_{i,2}$, ..., $v_{i,m}$. Comb product $K_4 \rhd I_G$ is shown in Figure 3, Figure 2 displays the BFS tree of I_G.

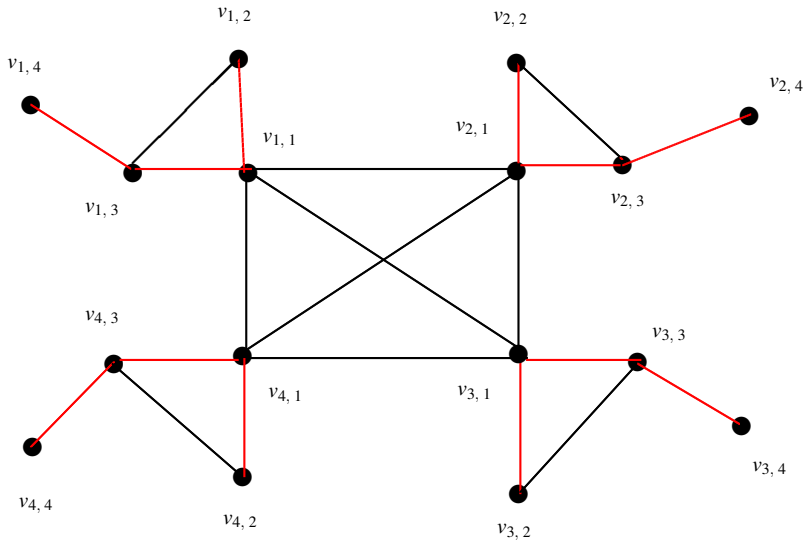

Figure 3. Comb product $K_4 \triangleright I_G$

Theorem 1 The $E_C(v)$ of each node v of $K_n \triangleright I_G$ is $\dfrac{1}{h+p+1}$, where v is situated at level p of the BFS tree T_{v_1} of the IG I_G.

Proof. Suppose v is any node of $K_n \triangleright I_G$. Again, let v is situated at level p, $0 \leq p \leq h$ on the BFS tree T_{v_1} of I_G (i^{th} copy) and h is the height of that tree. The vertex situated at level h of the BFS trees of I_G (other than i^{th} copy) is the farthest vertex from v. Now, the shortest distance from v to $v_{i,1}$ is p, and the distance between any pair of nodes of K_n is 1. So, the longest distance from v to other vertices of $K_n \triangleright I_G$ is $h+p+1$. Therefore, $E_C(v) = \dfrac{1}{max_{y \in V} d(v,y)} = \dfrac{1}{h+1+p}$. □

2.2 *Eccentricity centrality of $S_n \triangleright I_G$*

$S_n \triangleright I_G$ is the comb product between the star graph S_n of n vertices and I_G. Suppose that the grafting occurs at v_1 of each copy of I_G. Let the vertices of S_n and I_G be, respectively, $\{u_i : i = 1:1:n\}$ and $\{v_j : j = 1:1:m\}$. We re-label the nodes of $S_n \triangleright I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}, v_{i,2}, \ldots, v_{i,m}$. Figure 4 displays the graph $S_4 \triangleright I_G$.

Theorem 2 The eccentricity centrality $E_C(v)$ of each vertex v of $S_n \triangleright I_G$ is

$$E_C(v) = \begin{cases} \dfrac{1}{h+p+1}, & \text{if } v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of interval graph} \\ & \text{corresponding to the central vertex of } S_n, \\ \dfrac{1}{h+p+2}, & \text{if } v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of the interval graph} \\ & \text{corresponding to the non-central vertex of } S_n. \end{cases}$$

Proof. Let v be any vertex of $S_n \triangleright I_G$ as well as of I_G that is attached to the central node of S_n. Again let, v is situated at level p of the BFS tree T_{v_1} of that I_G and $h(T_{v_1}) = h$. The vertex situated at level h of T_{v_1} of I_G (attached with the non-central nodes of S_n) are the farthest vertex from v. The distance of $v_{1,1}$ from v is p, and the distance of $v_{i,1}$, $i = 2, 3, \ldots, n$ from $v_{1,1}$ is 1. So, the longest distance from v to other vertices of $S_n \triangleright I_G$ is $h+p+1$. Hence, $E_C(v) = \dfrac{1}{max_{y \in V} d(v,y)} = \dfrac{1}{h+1+p}$.

Again, let $v = v_{q,j}$ be any vertices of $S_n \triangleright I_G$ that are attached to the non-central vertex of the star graph. If v is situated at level p, then the distance from $v = v_{q,j}$ to $v_{q,1}$ is p. The distance from $v_{q,1}$ to $v_{i,1}$, $i = 2, \ldots, q-1, q+1, \ldots, n$ is

2. The vertices situated at level h of the BFS trees of I_G (attached with the other non-central vertices of S_n) are the farthest vertices from v. So, the longest distance from v is $h+p+2$. Therefore, $E_C(v) = \dfrac{1}{max_{y\in V} d(v,y)} = \dfrac{1}{h+2+p}$. □

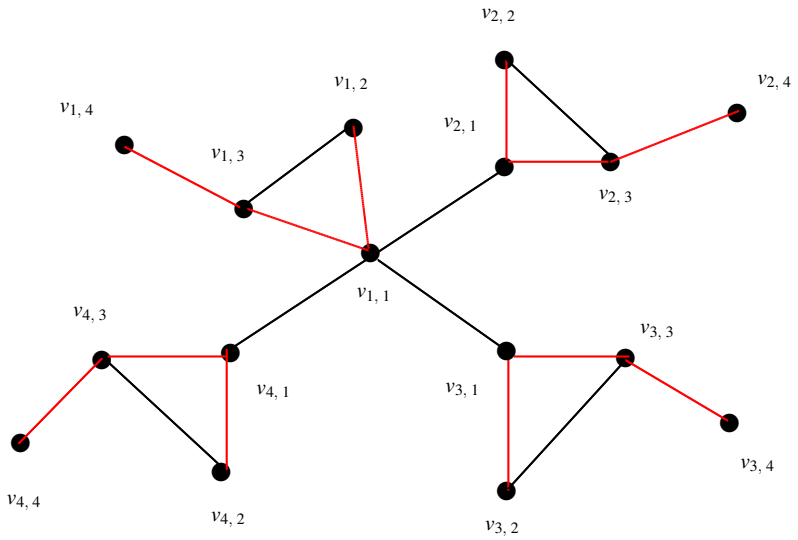

Figure 4. Comb product $S_4 \triangleright I_G$

2.3 *Eccentricity centrality of $W_n \triangleright I_G$*

$W_n \triangleright I_G$ is the comb product between the wheel graph W_n of $n+1$ vertices and the interval graph I_G. Suppose that the grafting occurs at v_1 of each copy of I_G. Let the vertices of W_n and I_G be, respectively, $\{u_i : i = 1:1:n+1\}$ and $\{v_j : j = 1:1:m\}$. We re-label the nodes of $W_n \triangleright I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}, v_{i,2}, \ldots, v_{i,m}$. Figure 5 displays the graph $W_4 \triangleright I_G$.

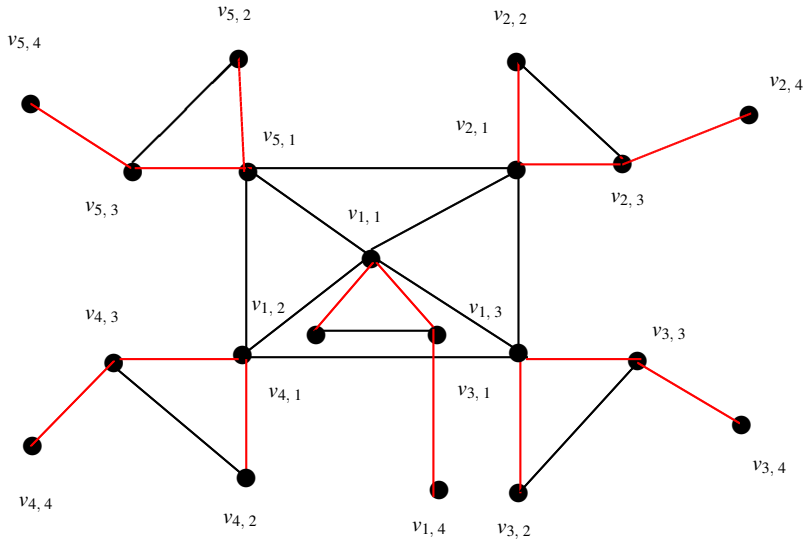

Figure 5. Comb product $W_4 \triangleright I_G$

Theorem 3 The eccentricity centrality $E_C(v)$ of each vertex v of $W_n \triangleright I_G$ is

$$E_C(v) = \begin{cases} \dfrac{1}{2+p+h}, & \text{if } v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of the interval graph} \\ & \text{corresponding to the non-central of } W_n, \\ \dfrac{1}{1+p+h}, & \text{if } v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of the interval graph} \\ & \text{corresponding to the center of } W_n. \end{cases}$$

Proof. Suppose v be any vertices of $W_n \triangleright I_G$ as well as of I_G that are attached to the central vertex of the wheel graph. Again, let v is situated at level p of the BFS tree T_{v_1} of that I_G and the height of the said BFS tree is h. The distance of $v_{1,1}$ from v is p, and the distance of $v_{i,1}$, $i = 2, 3, ..., n+1$ from $v_{1,1}$ is 1. The vertices situated at level h of the BFS trees of I_G (attached with the non-central vertices of W_n) is the farthest vertex from v. So, the longest distance from v to other vertices of $W_n \triangleright I_G$ is $h+p+1$. Hence, $E_C(v) = \dfrac{1}{max_{y \in V} d(v, y)} = \dfrac{1}{h+1+p}$.

Again, let $v = v_{q,j}$ be any vertices of $W_n \triangleright I_G$ that are attached to the non-central vertices of the wheel graph. If v is situated at level p of T_{v_1}, then the distance from $v = v_{q,j}$ to $v_{q,1}$ is p. The distance from $v_{q,1}$ to $v_{i,1}$, $i = 2, 3, ..., q-1, q+1, ..., n+1$ is 2. The vertices situated at level h of the BFS trees of I_G (attached with the other non-central vertices of W_n) are the farthest vertices from v. So, the longest distance from v is $h+p+2$. Hence, $E_C(v) = \dfrac{1}{max_{y \in V} d(v, y)} = \dfrac{1}{h+2+p}$. □

2.4 *Eccentricity centrality of comb product of cycle graph and interval graph*

$C_n \triangleright I_G$ is the comb product between the cycle graph C_n and the interval graph I_G. Suppose that the grafting occurs at v_1 of each copy of I_G. Let the vertices of C_n and I_G be, respectively, $\{u_i : i = 1 : 1 : n\}$ and $\{v_j : j = 1 : 1 : m\}$. We re-label the nodes of $C_n \triangleright I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}, v_{i,2}, ..., v_{i,m}$. Figure 6 displays the graph $C_4 \triangleright I_G$.

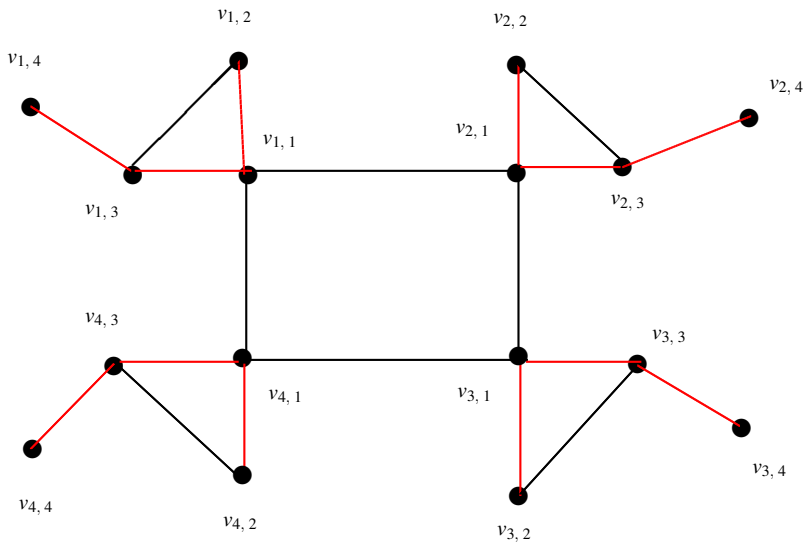

Figure 6. Comb product $C_4 \triangleright I_G$

Theorem 4 The eccentricity centrality $E_C(v)$ of each vertex v of $C_n \triangleright I_G$ is

$$E_C(v) = \begin{cases} \dfrac{1}{\frac{n}{2}+p+h}, & \text{if } n \text{ is even and } v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of interval graph,} \\ \dfrac{1}{\frac{n-1}{2}+p+h}, & \text{if } n \text{ is odd and } v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of interval graph.} \end{cases}$$

Proof. When n is even, let the vertices of C_n and I_G be $\left\{u_i : i = 1, 2, \ldots, \frac{n}{2}, \ldots, n\right\}$ and $\{v_j : j = 1 : 1 : m\}$, respectively. We re-label the nodes of $C_n \rhd I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}, v_{i,2}, \ldots, v_{i,m}$. Let $v = v_{i,j}$ be any vertices of $C_n \rhd I_G$. Also, let v is situated at level p of the BFS tree T_{v_1} of I_G (i^{th} copy) and $h(T_{v_1}) = h$. The distance from v to $v_{i,1}$ is p, and the greatest distance between two nodes of C_n is $\frac{n}{2}$. The vertices situated at level h of the BFS trees of I_G (attached with the vertex at a distance $\frac{n}{2}$ of C_n from $v_{i,1}$) is the farthest vertex from v. So, the longest distance from v to other vertices of $C_n \rhd I_G$ is $h + p + \frac{n}{2}$. Hence, $E_C(v) = \dfrac{1}{max_{y \in V} d(v, y)} = \dfrac{1}{h + \frac{n}{2} + p}$.

Again, when n is odd, let the vertices of C_n and I_G are $\left\{u_i : i = 1, 2, \ldots, \frac{n-1}{2}, \frac{n+1}{2}, ..., n\right\}$ and $\{v_j : j = 1 : 1 : m\}$, respectively. We re-label the nodes of $C_n \rhd I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}, v_{i,2}, \ldots, v_{i,m}$. Let $v = v_{i,j}$ be any vertices of $C_n \rhd I_G$. Also, let v is situated at level p of T_{v_1} of I_G (i^{th} copy) and $h(T_{v_1}) = h$. The distance from v to $v_{i,1}$ is p and the greatest distance between two nodes of C_n is $\frac{n+1}{2}$. The vertices be situated at level h of the tree T_{v_1} of I_G (attached with the node at a distance $\frac{n+1}{2}$ of C_n from $v_{i,1}$) is the farthest vertex from v. So, the longest distance from v to other vertices of $C_n \rhd I_G$ is $h + p + \frac{n+1}{2}$. Therefore, $E_C(v) = \dfrac{1}{max_{y \in V} d(v, y)} = \dfrac{1}{h + \frac{n+1}{2} + p}$. □

2.5 *Eccentricity centrality of $P_n \rhd I_G$*

$P_n \rhd I_G$ is the comb product between the path graph P_n and the interval graph I_G. Suppose that the grafting occurs at v_1 of each copy of I_G. Let the nodes of P_n and I_G be, respectively, $\{u_i : i = 1 : 1 : n\}$ and $\{v_j : j = 1 : 1 : m\}$. We re-label the nodes of $P_n \rhd I_G$ just as the nodes of the i^{th} copy of I_G are relabeled by $v_{i,1}, v_{i,2}, ..., v_{i,m}$. Figure 7 displays the graph $P_4 \rhd I_G$.

Theorem 5 The eccentricity centrality $E_C(v)$ of each vertex v of $P_n \rhd I_G$ is

$$E_C(v) = \begin{cases} \dfrac{1}{n-1+p+h}, & \text{if } v = u_{1,j} \text{ or } v = u_{n,j} \text{ and } v \text{ is situated at level } p \text{ of the} \\ & \text{BFS tree } T_{v_1} \text{ of the IG,} \\ \dfrac{1}{(n-i)+p+h}, & \text{if } v = u_{i,p}; n-i \geq i-1; i = 2 : 1 : n-1 \text{ and} \\ & v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of the IG,} \\ \dfrac{1}{(i-1)+p+h}, & \text{if } v = u_{i,p}; n-i < i-1; i = 2 : 1 : n-1 \text{ and} \\ & v \text{ is situated at level } p \text{ of the BFS tree } T_{v_1} \text{ of the IG.} \end{cases}$$

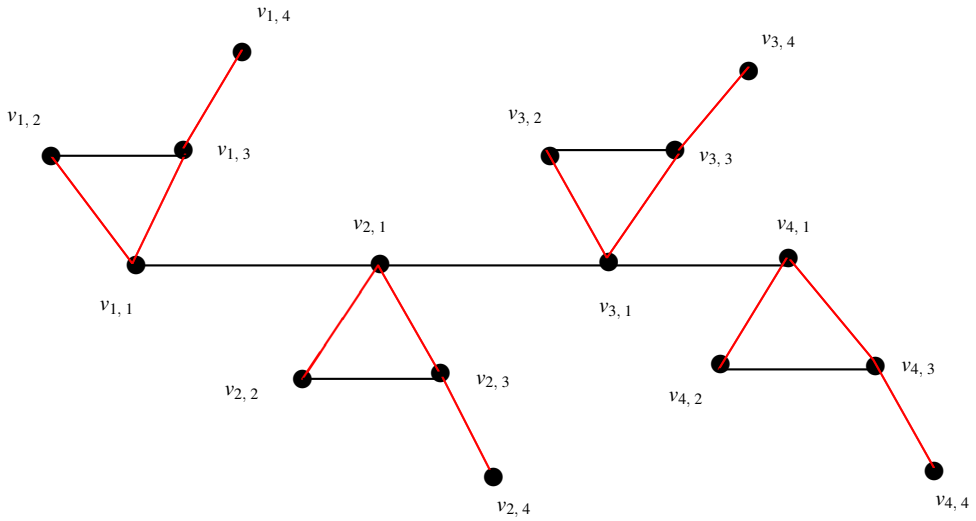

Figure 7. Comb product $P_4 \triangleright I_G$

Proof. Let $v = u_{1,j}$ be any vertices of $P_n \triangleright I_G$ that are attached to u_1 of the path graph P_n. Again, let v is situated at level p of the BFS tree T_{v_1} of I_G and $h(T_{v_1}) = h$. The distance of $u_{1,1}$ from v is p, and the distance of $u_{n,1}$ from $u_{1,1}$ is $n-1$. The nodes situated at level h of the tree T_{v_1} of I_G (attached with u_n of P_n) is the farthest node from v. So, the longest distance from v to other vertices of $P_n \triangleright I_G$ is $h+p+n-1$. Hence, $E_C(v) = \dfrac{1}{max_{y\in V} d(v,y)} = \dfrac{1}{h+n-1+p}$. If $v = u_{n,j}$, then we get the same result.

Again, let $v = u_{i,p}$ be any vertices of $P_n \triangleright I_G$ that are attached to u_i of the path graph, i.e., v is situated at level p of the tree T_{v_1} of I_G. The distance from $v = v_{i,p}$ to $v_{i,1}$ is p. If $n-i \geq i-1$, then the vertex situated at level h of T_{v_1} of I_G (attached with u_{n-i} of P_n) is the farthest vertices from v. So, the longest distance from v is $h+p+n-i$. Hence, $E_C(v) = \dfrac{1}{max_{y\in V} d(v,y)} = \dfrac{1}{h+p+n-i}$.

Again, if $n-i < i-1$, then the vertex situated at level h of the tree T_{v_1} of I_G (attached with u_{i-1} of P_n) is the farthest vertex from v. So, the longest distance from v is $h+p+i-1$. In this case, $E_C(v) = \dfrac{1}{max_{y\in V} d(v,y)} = \dfrac{1}{h+p+i-1}$. □

2.6 *Eccentricity centrality of comb product between two interval graphs*

$I_H \triangleright I_G$ is the comb product between two interval graphs I_H and I_G. Suppose that the grafting occurs at v_1 of each copy of I_G. Let the nodes of I_H and I_G be, respectively, $\{u_i : i = 1:1:n\}$ and $\{v_j : j = 1:1:m\}$. We re-label the nodes of $I_H \triangleright I_G$ as $\{x_{j,q}(i,\ p) : i = 1:1:n;\ j = 1:1:m\}$, where p and q indicate the level of the BFS tree of I_H and I_G, respectively. Interval graph I_H and its representation and the BFS trees with root at u_1 and u_4 are, respectively, shown in Figure 8, Figure 9, and Figure 10. The comb product of two interval graphs $I_H \triangleright I_G$ is shown in Figure 11. To find the eccentricity centrality of the comb product between two interval graphs, we first find the eccentricity of the interval graph I_H.

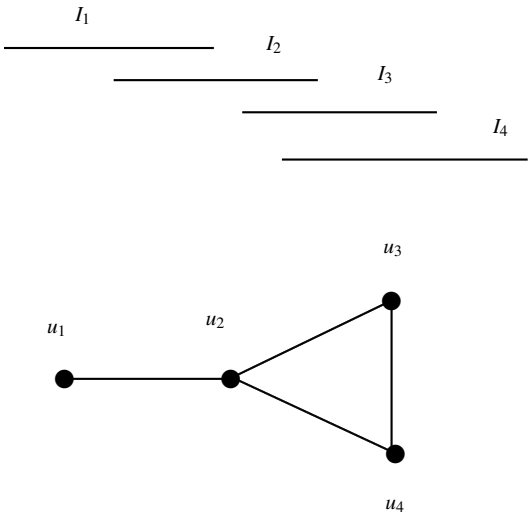

Figure 8. Interval graph I_H and its representation

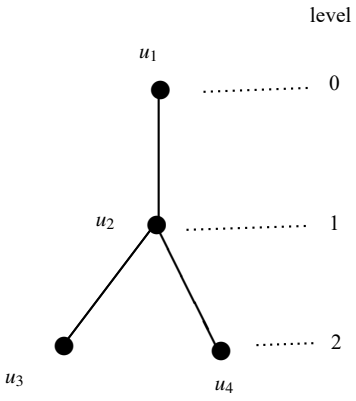

Figure 9. BFS tree T_{u_1} of I_H

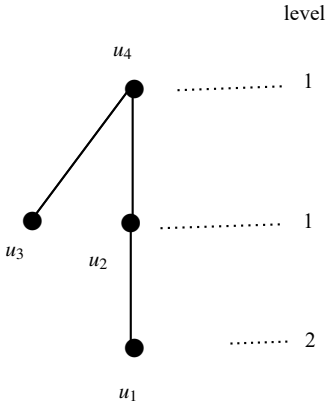

Figure 10. BFS tree $T_{u_{lmax}}$ of I_H, where $u_{lmax} = u_4$

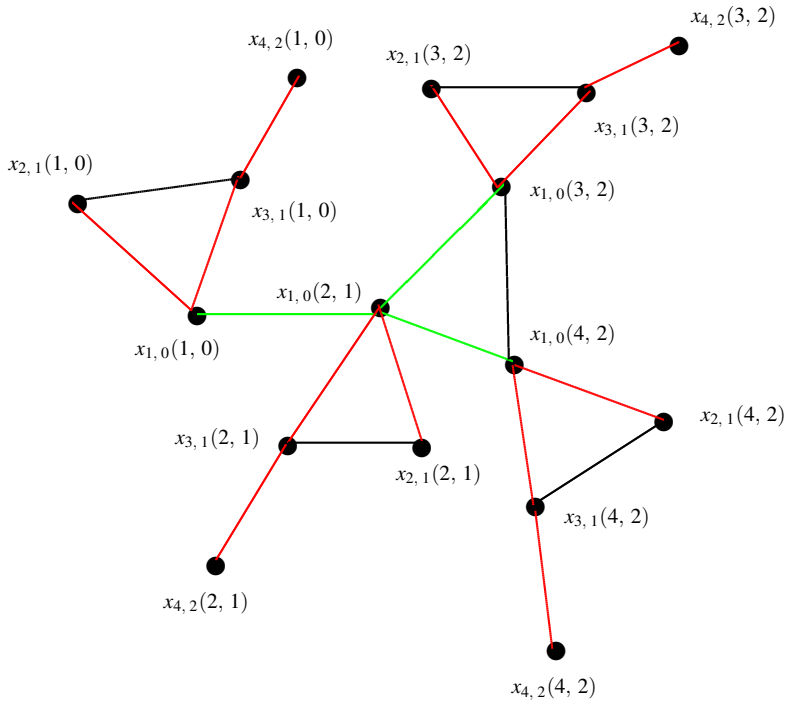

Figure 11. Comb product $I_H \triangleright I_G$

2.6.1 *BFS tree of interval graphs*

Here we present a new faster algorithm to make a BFS tree T_{u_x}, u_x is the root, of IG.

Algorithm IBFS

Input: An interval representation $[a_i, b_i]$, $i = 1 : 1 : n$.

Output: The BFS tree T_{u_x} with root as u_x.

Step 1: Select and mark the interval I_x in IR and set the root as u_x.

Step 2: Scan the interval $[a_x, b_x]$

Step 2.1: Find the sets $A = \{a_i : a_i$ of all unmarked intervals whose a_i or b_i are in $[a_x, b_x]\}$,
$B = \{b_i : b_i$ of all unmarked intervals whose a_i or b_i are in $[a_x, b_x]\}$ and marked these intervals.

Step 2.2: Set $S = \{i : a_i, b_i \in A \cup B\}$.

Step 2.3: Set $parent(u_k) = u_x$ for all $k \in S$.

Step 2.4: Compute m, M, where m is the subscript of $a_i \in A$ with left-most position in IR and $M = max\{i : b_i \in B\}$.

Step 3: {Right scanning}

Step 3.1: Scan all unmarked intervals whose $a_i \in [r_l, r_r] = [b_x, b_M]$ and make a set R that contains all the b_i corresponding to these intervals.

Step 3.2: If R is the empty set, then move to Step 4, else move to the next Step.

Step 3.3: Set $parent(u_k) = u_M$ for all $k \in B = \{i : b_i \in R\}$.

Step 3.4: Reset the interval $[r_l, r_r]$ for the next iteration such that $r_l = r_r$, $r_r = b_M$, where $M = maxB$.

Step 4: {Left scanning}

Step 4.1: Scan all unmarked intervals whose $b_i \in [l_l, l_r] = [a_m, a_x]$ and make a set L that contains all a_i corresponding to these intervals.

Step 4.2: If both L and R are empty sets, then stop.
Else if L is the empty set, then move to Step 3.
Else move to the next Step.

End if.

Step 4.3: Set $parent(u_k) = u_m$ for all $k \in A = \{i : a_i \in L\}$.

Step 4.4: Reset the interval $[l_l, l_r]$ for next iteration such that $l_r = l_l$, $l_l = a_m$, where m is the subscript of $a_i \in L$ with left most position in IR.

Step 4.5: Move to Step 3.

Stop

end IBFS.

From the above Algorithm IBFS, we get a set of $parents(u_k)$, $k \in A \cup B \cup S$, by which we can construct the BFS tree T_{u_x}. If we use the proposed Algorithm IBFS for the IG I_H displayed in Figure 8 and take u_1 as the root, then $parent(u_2) = u_1$, $parent(3) = u_2$ and $parent(u_4) = u_2$. Finally, we can easily construct the BFS tree T_{u_1} as shown in Figure 9.

For the tree T_{u_1}, we consider $u_{lmax} \to parent(u_{lmax} \to \cdots \to u_1)$ as the axial path of it. So, the height of the tree T_{u_1} is the length of axis.

Theorem 6 Algorithm IBFS executes in $O(n)$ time.

Proof. In Algorithm IBFS, Step 1 needs constant time. To compile Step 2, we require $O(n)$ time. In Step 3, at each iteration during the right scanning we scan only unmarked intervals. So, to finish the right scanning we need only $O(n)$ time. Also, in Step 4, at each iteration during the left scanning we scan only unmarked intervals. So, to finish the left scanning, we need only $O(n)$ time. Therefore, the algorithm IBFS executes in $O(n)$ time. □

2.6.2 *Some notations*

Here we present some notations (Table 1).

Table 1. Notations

Symbol	Description
$e(v)$	Eccentricity of the node v.
T_x	BFS tree with root as x.
IG	Interval Graph.
I_n	IG with n vertices.
IR	Interval Representation of IG.
u_{lmax}	Vertex whose corresponding interval's left endpoint's position is maximum in IR.
$h(T)$	Height of the tree T.
u_i^*	The vertex at level i on the axial path of T_{u_1}.
A_i	Set of nodes at level i of T_{u_1} excluding u_i^*.
u_i'	The vertex at level i on the axial path of $T_{u_{lmax}}$.
A_i'	Set of nodes at level i of $T_{u_{lmax}}$ excluding u_i'.
h	$h = h(T_{u_1}) = h(T_{u_{lmax}})$.

2.6.3 *Eccentricity of IG*

The eccentricity of a node v of V, we denote it by $e(v)$, represents the greatest length among all shortest paths starting from v and finishing at other remaining nodes, i.e., $e(v) = Max\{d(v, u_x) : u_x \in V\}$. If a graph is disconnected (may have isolated vertices), the eccentricity of any vertex is infinite or undefined. For this reason, we consider here a simple, undirected, and connected interval graph. To find the eccentricity of all vertices of interval graphs, we state and prove some important results, and then we also design an algorithm.

First, we call a vital result on interval graphs.

Lemma 1 If a, b, and c are any three nodes of an IG, where $a < b < c$ and a is directly connected with c then $(b, c) \in E$

Now, we present some new results.

Lemma 2 The vertex u_{lmax} lies at the last level of the tree T_{u_1}.

Proof. Let us assume that the node u_{lmax} is not at the last level of the tree T_{u_1}. This means the interval corresponding to the node u_{lmax} is not scanned in the last interval $[r_l, r_r]$. That means a_{lmax} is not at the maximum position in IR. Which is a contradiction as per the definition of u_{lmax}. Therefore, u_{lmax} is at the last level of the tree T_{u_1}. □

Lemma 3 For the tree T_{u_1}, $(u, u_1^*) \in E$, for all u in A_1.

Proof. According to the scanning procedure of the Algorithm IBFS, b_i (corresponding to u_1^*) $> b_j$ (corresponding to the vertex $u_j \in A$). So, $u_1 < u_j < u_i$, where $1 < j < i$. Therefore, by Lemma 1, $(u, u_1^*) \in E$, for all u in A_1. □

Lemma 4 For an interval graph G, $h(T_{u_1}) = diameter(G)$.

Proof. As u_1 is at level 0 and u_{lmax} is at level h of the tree T_{u_1} of the IG G, so, by Lemma 3, $h(T_{u_1}) = diameter(G)$. □

Lemma 5 For the tree $T_{u_{lmax}}$, $(u, u_1') \in E$, for all u in A_1'.

Proof. It can be proved in the same way as Lemma 3. □

Lemma 6 For IG G, $h(T_{u_{lmax}}) = diameter(G)$.

Proof. It can be proved in the same way as Lemma 4. □

Lemma 7 For IG G, $h(T_{u_1}) = h(T_{u_{lmax}})$.

Proof. This result can be easily verified by comparing the results of the Lemma 4, 6. □

Lemma 8 If the node v is situated at level p on the BFS tree T_{u_1}, then v lies at level $h-p$ or $h-p+1$ of the BFS tree $T_{u_{lmax}}$.

Proof. We know $h = h(T_{u_1}) = h(T_{u_{lmax}})$ and the BFS trees T_{u_1} and T_{ulmax} are constructed by the same Algorithm IBFS and the position of the right endpoint of u_{lmax} is maximum among all a_i, $i = 1:1:n$ in IR. Now, if the node v is at level p of T_{u_1}, then the interval corresponding to v is scanned during the $(p-1)$th right scanning during the construction of T_{u_1}. So, the interval corresponding to v shall be scanned during the $(h-p-1)$th or $(h-p)$th right scanning during the construction of T_{u_1}. This implies that v lies at level $h-p$ or $h-p+1$ of the BFS tree $T_{u_{lmax}}$. □

Lemma 9 If v is the root of either T_{u_1} or $T_{u_{lmax}}$ and the height of these BFS trees is h, then the eccentricity of v is h.

Proof. Let v be the root of the tree T_{u_1}, and $h(T_{u_1}) = h$. The vertex/vertices at level h is/are the farthest vertex from v and the maximum distance is h. So by definition, the eccentricity of v is h. Again, if we consider v as the root of $T_{u_{lmax}}$ then we will get the same result. □

Lemma 10 If the height of the BFS tree T_{u_1} is even and v is situated at level p where $p > \frac{h}{2}$ then the eccentricity of v is p.

Proof. Let the height h of the BFS tree T_{u_1} be even, and v is situated at level p where $p > \frac{h}{2}$. So, $p > h-p$. The vertex/vertices at level p is/are the farthest vertex from u_1. So, the longest distance from node v to other nodes is p. So, by definition, the eccentricity of v is p. □

Lemma 11 If the height of the BFS tree T_{u_1} be even, and v is situated at level p on T_{u_1} as well as q of $T_{u_{lmax}}$, $p < \frac{h}{2}$ then the eccentricity of v is q.

Proof. Let the height of the BFS tree T_{u_1} is even and v is situated at level p where $p < \frac{h}{2}$. So, by Lemma 3, v is situated at level $h-p$ or $h-p+1$ on the BFS tree $T_{u_{lmax}}$. As $p < \frac{h}{2}$, the distance from v to u_1 or the vertices at level 1 is less than the distance from v to the vertices at level h of T_{u_1}, $p < h-p$. So, the vertex/vertices at level h of T_{u_1} is/are the furthest vertex/vertices from v. But, the distance from v to those vertices is either $h-p$ or $h-p+1$. For this ambiguous case, we compute the level of the vertex v on $T_{u_{lmax}}$. If v is situated at level q of $T_{u_{lmax}}$ then, $e(v) = q$ as $q = h-p > p$ or $q = h-p+1 > p$. □

Lemma 12 If the height of the BFS tree T_{u_1} is even and v is situated on the axial path at level p of T_{u_1} where $p = \frac{h}{2}$ then the eccentricity of v is p.

Proof. Let the height of the BFS tree T_{u_1} be even, and v is situated on the axial path at level p of T_{u_1} where $p = \frac{h}{2}$. The distance from the node v to the node(s) situated at level h is $h - p = h - \frac{h}{2} = \frac{h}{2}$ and $d(v, u_1) = \frac{h}{2}$. The distance from the node v to other nodes of G is $\leq \frac{h}{2}$. Therefore, the longest distance is $\frac{h}{2} = p$. Hence, the eccentricity of v is p. □

Lemma 13 If the height h of the BFS tree T_{u_1} is even and v is a leaf node, situated at level $\frac{h}{2}$ of T_{u_1} then

$$e(v) = \begin{cases} \frac{h}{2}+2, & \text{if } v \text{ is not connected with the vertices at level } \frac{h}{2}+1 \text{ and the node on} \\ & \text{axial path at level } \frac{h}{2} \text{ of } T_{u_1} \\ \frac{h}{2}+1, & \text{if } v \text{ is not connected with the vertices at level } \frac{h}{2}+1 \text{ and conncted with} \\ & \text{the node on axial path at level } \frac{h}{2} \text{ of } T_{u_1} \\ \frac{h}{2}, & \text{otherwise} \end{cases}$$

Proof. Let the height h of the BFS tree T_{u_1} be even, and v is a leaf node and situated at level $\frac{h}{2}$ on T_{u_1}. Again, let the vertices of the axial path be u_i where $i = 0, 1, 2, .., \frac{h}{2}, ..., h$ and u_h^* be any leaf node of T_{u_1} at level h.

Case 1:

Let v be not adjacent to the vertices at level $\frac{h}{2}+1$ and the node on the axial path at level $\frac{h}{2}$ of T_{u_1}. So, there is only one path from v to the vertices at level h ($v \to u_{\frac{h}{2}-1} \to u_{\frac{h}{2}} \to u_h/u_h^*$). Therefore, the distance from v to the vertices at level h is $h - \frac{h}{2} + 2 = \frac{h}{2} + 2$. Also, $d(v, u_1) = \frac{h}{2}$. Therefore, $Max\{d(u, v) : u \in V\} = \frac{h}{2} + 2$. Hence $e(v) = \frac{h}{2} + 2$.

Case 2:

Let v be not adjacent to the vertices at level $\frac{h}{2}+1$ and adjacent to the vertex on the axial path at level $\frac{h}{2}$ of T_{u_1}. The shortest path between v and the vertices at level h is $v \to u_{\frac{h}{2}} \to u_{\frac{h}{2}+1} \to u_h/u_h^*$. Therefore, the distance from v to the vertices at level h is $h - \frac{h}{2} + 1 = \frac{h}{2} + 1$. Also, $d(v, u_1) = \frac{h}{2}$. So, $Max\{d(u, v): u \in V\} = \frac{h}{2} + 1$. Hence, $e(v) = \frac{h}{2} + 1$.

Case 3:

Let v be adjacent to the vertices on the axial path at level $\frac{h}{2}+1$ of T_{u_1}. The shortest path between v and the nodes at level h is $v \to u_{\frac{h}{2}+1} \to u_{\frac{h}{2}+2} \to u_h/u_h^*$. Therefore, the distance from v to the vertices at level h is $h - \frac{h}{2} = \frac{h}{2}$. Also, $d(v, u_1) = \frac{h}{2}$. Therefore, $Max\{d(u, v) : u \in V\} = \frac{h}{2}$. Hence $e(v) = \frac{h}{2}$. □

Lemma 14 If the height of the BFS tree T_{u_1} is odd and v is situated at level p where $p > \frac{h+1}{2}$ then the eccentricity of v is p.

Proof. Let the height h of the BFS tree T_{u_1} be odd, and v is situated at level p where $p > \frac{h+1}{2}$. As $p > \frac{h+1}{2}$, then the vertex/vertices at level p is/are the farthest vertex from u_1. Also, the vertices at level h are at a distance at most $h - p + 2$ from v, as $p > \frac{h+1}{2}$. So, $Max\{d(u, v) : u \in V\} = p$. Therefore, by definition, the eccentricity of v is p. □

Lemma 15 If the height of the BFS tree T_{u_1} is odd and v is situated at level p on T_{u_1} as well as at level q on $T_{u_{lmax}}$, $p < \frac{h+1}{2}$ then the eccentricity of v is q.

Proof. Let the height of the BFS tree T_{u_1} be odd, and v is situated at level p where $p < \frac{h+1}{2}$. So, by Lemma 3, v is situated at level $h-p$ or $h-p+1$ on the BFS tree $T_{u_{lmax}}$. As $p < \frac{h+1}{2}$, the distance from v to u_1 or the vertices at level 1 p, which is less than the distance from v to the vertices at level h of T_{u_1}, i.e., the vertex/vertices at level h of T_{u_1} is/are the furthest vertex/vertices from v. But, the distance from v to those vertices is either $h-p$ or $h-p+1$. For this ambiguous case, we find the level of the vertex v on $T_{u_{lmax}}$. If v is situated at level q of $T_{u_{lmax}}$ then $e(v) = q$. □

Lemma 16 If the height of the BFS tree T_{u_1} is odd and v is situated on the axial path at level p of T_{u_1} where $p = \frac{h+1}{2}$ then the eccentricity of v is p.

Proof. Let the height of the BFS tree T_{u_1} be odd, and v is situated on the axial path of T_{u_1} at level p where $p = \frac{h+1}{2}$. The distance of the nodes situated at level h from the node v is $h-p = h - \frac{h+1}{2} = \frac{h-1}{2}$, and the distance of v from u_1 is $\frac{h+1}{2}$. So, $d(v,u) \leq \frac{h+1}{2}, \forall u \in V(G)$. Therefore, the maximum distance is $\frac{h+1}{2} = p$. Hence $e(v) = p$. □

Lemma 17 If the height of the BFS tree T_{u_1} is odd and v is a leaf node, situated at level p of T_{u_1} where $p = \frac{h+1}{2}$ then

$$e(v) = \begin{cases} h-p+2, & \text{if } v \text{ is not connected with the nodes at level } p+1 \text{ and the node on axial path at level } p \text{ of } T_{u_1} \\ p, & \text{otherwise} \end{cases}$$

Proof. Let the height h of the BFS tree T_{u_1} be odd, and v is a leaf node situated at level p of T_{u_1} where $p = \frac{h+1}{2}$. Again, let the vertices of the axial path be u_i, where $i = 0, 1, 2, .., p, ..., h$ and the leaf node be u_{ij}, where i is the level of v and j is the original subscript of v.

Case 1:

Let v be not adjacent to the vertices at level $p+1$ and the node on the axial path at level p of T_{u_1}. So, there is only one path from v to the vertices at level h ($v \to u_{p-1} \to u_p \to u_h/u_{hj}$). Therefore, the distance from v to the vertices at level h is $h-p+2 = h - \frac{h+1}{2} + 2 = \frac{h+3}{2} = \frac{h+1}{2} + 1$. The distance from v to u_1 is $p = \frac{h+1}{2}$. Therefore, the maximum distance is $h-p+2$. Hence $e(v) = h-p+2$.

Case 2:

Let, v be adjacent to the vertex at level $p+1$ or adjacent to the node on the axial path at level p of T_{u_1}. If v is adjacent to the vertex at level $p+1$ then the shortest path between v and the vertices at level h is $v \to u_{p+1} \to u_{p+2} \to u_h/u_{hj}$. Therefore, the distance from v to the vertices at level h is $h-p = h - \frac{h+1}{2} = \frac{h-1}{2}$. The distance from v to u_1 is $p = \frac{h+1}{2}$. Therefore, the maximum distance is p. Hence $e(v) = p$.

If v is adjacent to the node at level p on the axial path, then the shortest path between v and the vertices at level h is $v \to u_p \to u_{p+1} \to u_h/u_{hj}$. Therefore, the distance from v to the vertices at level h is $h-p+1 = h - \frac{h+1}{2} + 1 = \frac{h+1}{2} = p$. The distance from v to u_1 is $p = \frac{h+1}{2}$. Therefore, the maximum distance is p. Hence $e(v) = p$. □

2.6.4 *Algorithm and complexity*

Here, we are ready to propose a faster algorithm to obtain the eccentricity of all vertices of IG G.

Algorithm ECCEN-INT

Input: Interval representation I_i, $i = 1:1:n$ of the IG G.

Output: Eccentricity of each node u_i, $i = 1:1:n$ of the IG G.

Step 1: Construct two BFS-trees T_{u_1} and T_{u_n} and compute the height h.
Step 2: Compute the level of u_i (denoted by p_i) for T_{u_1} for $i = 1:1:n$.
Step 3: Compute the level of u_i (denoted by q_i) for $T_{u_{lmax}}$ for $i = 1:1:n$.
Step 4: for $i = 1:1:n$

If h is even and $p_i > \frac{h}{2}$, then $e(u_i) = p_i$. //Lemma 10

Else if h is even and $p_i < \frac{h}{2}$, then compute q_i and $e(u_i) = q_i$. //Lemma 11

Else if h is even and u_i is situated on the axial path at level p_i of T_{u_1} where $p_i = \frac{h}{2}$, then $e(u_i) = p_i$. //Lemma 12

Else if h is even and u_i is leaf node, situated at level p_i of T_{u_1} where $p_i = \frac{h}{2}$, then (by Lemma 13)

$$e(u_i) = \begin{cases} \frac{h}{2}+2, & \text{if } u_i \text{ is not connected with the nodes at level } \frac{h}{2}+1 \text{ and the node on} \\ & \text{axial path at level } \frac{h}{2} \text{ of } T_{u_1} \\ \frac{h}{2}+1, & \text{if } u_i \text{ is not connected with the nodes at level } \frac{h}{2}+1 \text{ and conncted with} \\ & \text{the node on axial path at level } \frac{h}{2} \text{ of } T_{u_1} \\ \frac{h}{2}, & \text{otherwise} \end{cases}$$

Else if h is odd and $p_i > \frac{h+1}{2}$, then $e(u_i) = p_i$. //Lemma 14

Else if h is odd and $p_i < \frac{h+1}{2}$, then compute q_i and $e(u_i) = q_i$. //Lemma 15

Else if h is odd and u_i is situated on the axial path at level p_i of T_{u_1} where $p_i = \frac{h+1}{2}$, then $e(u_i) = p_i$ (by Lemma 16)

Else if h is odd and u_i is a leaf node, situated at level p_i of T_{u_1}, where $p_i = \frac{h+1}{2}$, then (by Lemma 17)

$$e(u_i) = \begin{cases} h-p_i+2, & \text{if } v \text{ is not connected with the nodes at level } p_i+1 \text{ and the node on} \\ & \text{axial path at level } p_i \text{ of } T_{u_1} \\ p_i, & \text{otherwise} \end{cases}$$

End if
End
End ECCEN-INT.

Theorem 7 Algorithm ECCEN-INT takes $O(n)$ time for computing eccentricity of all nodes of IG G .

Proof. In 1st Step, for constructing two BFS-trees T_{u_1} and $T_{u_{lmax}}$, and computing the height h, $O(n)$ time is necessary. The second and third Steps need $O(n)$ time to compute p_i and q_i for each u_i, separately. Step 4 takes $O(n)$ time to calculate the eccentricity of each vertex of G. Hence, the Algorithm ECCEN-INT takes $O(n)$ time. □

Note 1: Our ECCEN-INT algorithm has been implemented with standard BFS operations that maintain integer vertex levels using well-defined queue-based traversal. For practical values of n, the computation of vertex levels does not lead to numerical or overflow errors, as the levels are represented by integer indices corresponding to graph vertices. For

extremely large graphs, that is, for very large values of n, appropriate data types (such as long integers) should be used to avoid potential computational inaccuracies when storing vertex levels in the following way.

• Since the maximum level in a BFS tree is at most the diameter of the graph, it will usually be much smaller than n.

• Instead of using absolute numbers for vertex levels, store only relative levels (for example, starting from 0 in each BFS traversal). We have followed this rule to level the vertices of BFS tree of interval graphs.

• Instead of storing levels for all vertices as large integers, store them in an array of normal integers (32-bit).

Theorem 8 The eccentricity centrality $E_C(v)$ of each vertex v of $I_H \rhd I_G$ is $\frac{1}{e(u)+q+h_2}$ where $v = x_{j,q}(i,\ p)$, $u = x_{1,0}(i,\ p)$, $i = 1:1:n,\ j = 1:1:m$ and h_1 and h_2 are the height of the BFS tree of I_H with root as u_1 and I_G with root as v_1 respectively.

Proof. Let $v = x_{j,q}(i,\ p)$ be any vertices of $I_H \rhd I_G$ that attached with I_H where p and q indicate the level of the BFS tree of I_H and I_G. Again let, the height of the said BFS trees I_H and I_G are h_1 and h_2, respectively. The distance from v to $x_{1,0}(i,\ p)$ is q. Also, the maximum distance from $x_{1,0}(i,\ p)(= u)$ to other vertices of I_H is the eccentricity of u, is $e(u)$. The vertices situated at level h_2 of the BFS trees of I_G is the farthest vertex from v. So, the longest distance from v to other vertices of $I_H \rhd I_G$ is $q + e(u) + h_2 = e(u) + q + h_2$. Hence, the eccentricity centrality of v is $E_C(v) = \frac{1}{max_{y\in V} d(v,y)} = \frac{1}{e(u)+q+h_2}$. □

Illustrative example: We consider the graph $I_H \rhd I_G$ displayed in Figure 11. First, we apply the Algorithm ECCENINT to find the eccentricity of all vertices of I_H, and then we apply the result of Theorem 7. For I_H, we have $e(u_1) = 2$, $e(u_2) = 1$, $e(u_3) = 2$, $e(u_4) = 2$. The height of the BFS tree corresponding to I_G is $h_2 = 2$. For this comb product graph, $u = x_{1,0}(i,\ p)$, $v = x_{j,q}(i,\ p)$, $i,\ j = 1,\ 2,\ 3,\ 4$ and $p,\ q = 0,\ 1,\ 2$.

Now we apply the formula, we have

$$E_C(x_{1,0}(1,\ 0)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+0+2} = \frac{1}{4}.$$

$$E_C(x_{2,1}(1,\ 0)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+1+2} = \frac{1}{5}.$$

$$E_C(x_{3,1}(1,\ 0)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+1+2} = \frac{1}{5}.$$

$$E_C(x_{4,2}(1,\ 0)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+2+2} = \frac{1}{6}.$$

$$E_C(x_{1,0}(2,\ 1)) = \frac{1}{e(u)+q+h_2} = \frac{1}{1+0+2} = \frac{1}{3}.$$

$$E_C(x_{2,1}(2,\ 1)) = \frac{1}{e(u)+q+h_2} = \frac{1}{1+1+2} = \frac{1}{4}.$$

$$E_C(x_{3,1}(2,\ 1)) = \frac{1}{e(u)+q+h_2} = \frac{1}{1+1+2} = \frac{1}{4}.$$

$$E_C(x_{4,2}(2,\ 1)) = \frac{1}{e(u)+q+h_2} = \frac{1}{1+2+2} = \frac{1}{5}.$$

$$E_C(x_{1,0}(3,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+0+2} = \frac{1}{4}.$$

$$E_C(x_{2,1}(3,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+1+2} = \frac{1}{5}.$$

$$E_C(x_{3,1}(3,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+1+2} = \frac{1}{5}.$$

$$E_C(x_{4,2}(3,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+2+2} = \frac{1}{6}.$$

$$E_C(x_{1,0}(4,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+0+2} = \frac{1}{4}.$$

$$E_C(x_{2,1}(4,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+1+2} = \frac{1}{5}.$$

$$E_C(x_{3,1}(4,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+1+2} = \frac{1}{5}.$$

$$E_C(x_{4,2}(4,2)) = \frac{1}{e(u)+q+h_2} = \frac{1}{2+2+2} = \frac{1}{6}.$$

The above results satisfy the graph we considered.

2.6.5 *Comparative analysis of algorithms for calculating eccentricity in a interval graphs*

Here, we present a comparative analysis of existing algorithms for calculating eccentricity in interval graphs, focusing on methods, authors, and time complexity.

Table 2. Comparative chart

Author(s)	Method time	Complexity
Gavril et al. [17]	Structural properties	$O(n^2)$
Olariu [18]	BFS	$O(m+n)$
Nandi et al. [15]	BFS and our studied result	$O(n)$ (our result)

From Table 2, we see that our algorithm is more efficient than the existing results.

3. Application of the eccentricity centrality of $P_n \rhd I_m$

Although the Cartesian product of two graphs is commonly used to model a warehouse network, in this study, we attempt to represent this network through the comb of two graphs, as it also captures the hierarchical and distributed nature of the supply chain. Here, we present a real application of the comb product between P_n and IG I_m. Details are given below in stepwise.

Step 1: Problem and objectives.

Problem: Suppose an online product-selling company with several warehouses distributed across different districts, some of which serve as main warehouses while others are local warehouses. These main warehouses are connected to each other by logistic support. Each main warehouse is also connected by logistic support with local warehouses in these districts. Suppose this company operates in three districts, say A (Howrah), B (Purba Medinipur), and C (Paschim Medinipur). The company has one main warehouse (situated near rail station/national highway) and five local warehouses (situated in the town area) in each district.

Objective: To determine the most centrally located warehouse that minimizes the longest path to other warehouses in this network.

Step 2: Graph model for the warehouse network.

To model problem, we use the comb product of a path graph and an interval graph. We can approach it as follows. The path graph will represent the main warehouses, while the interval graphs will represent local warehouses within these main warehouses in each district.

Step 2.1: Construction of a path graph for the main warehouses.

Here, we consider that the main warehouses are connected to each other in a line. So they will form a path graph. For example, if X (Howrah), Y (Mecheda), and Z (Kharagpur) are, respectively, the main warehouses in the districts A, B, and C , then the path graph P_3 will be like Figure 12.

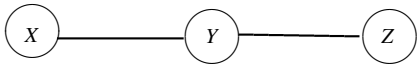

Figure 12. Path graph P_3

Step 2.2: Construction of interval graphs for local warehouses in each district.

In each district, we construct one interval graph that represents the interconnection or overlapping among the local warehouses. Each node in this interval graph represents a local warehouse. The length of each interval in an interval graph represents the service range (logistical reach of a warehouse) of a local warehouse. This means that a longer interval would indicate a larger service area, meaning the warehouse can serve customers or connect with other warehouses over a greater distance. A shorter interval would indicate a smaller or more localized service area. Two nodes of an interval are connected if their service areas overlap. So, we construct three interval graphs for three districts. Let the local warehouses in three districts be as follows.

In district A: M (Mourigram), U (Uluberia), B (Bagnan), A (Amta), and U^* (Udaynarayanpur).

In district B: P (Panskura), T (Tamluk), N (Nandakumar), K (Kanthi), and E (Egra).

In district C: M^* (Midnapore), S (Salboni), G (Garhbeta), C^* (Chadrakona), and G^* (Ghatal).

We assume that these three interval graphs are isomorphic, only the vertex labeling is distinct. For convenience, let I_5^A, I_5^B, I_5^C (displayed in Figure 13) are respectively the interval graphs in the districts A, B, and C. Let these three interval graphs be isomorphic to I_5.

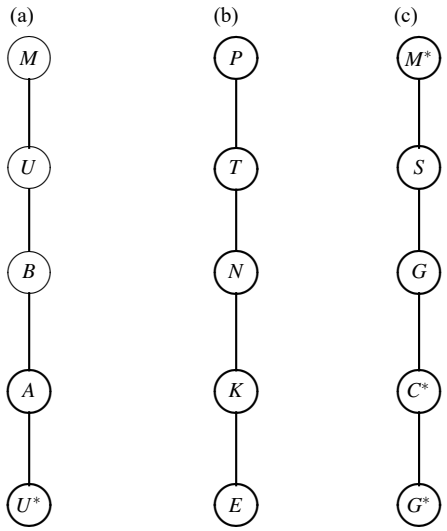

Figure 13. (a) I_5^A, (b) I_5^B, (c) I_5^C

Step 3: Construction of the comb product graph.

The comb product $P_3 \triangleright I_5$ between a complete graph P_3 and an interval graph I_5 involves connecting each vertex in P_3 to a copy of I_5. This structure will allow for both robust central connectivity among main warehouses and local connectivity within three districts, A, B, and C. Let the vertices of $P_3 \triangleright I_5$ (Figure 14) be $v_{i,j}$, $i = 1, 2, 3$; $j = 1, 2, 3, 4, 5$. Its real image (drawn by Google map) is drawn in Figure 15.

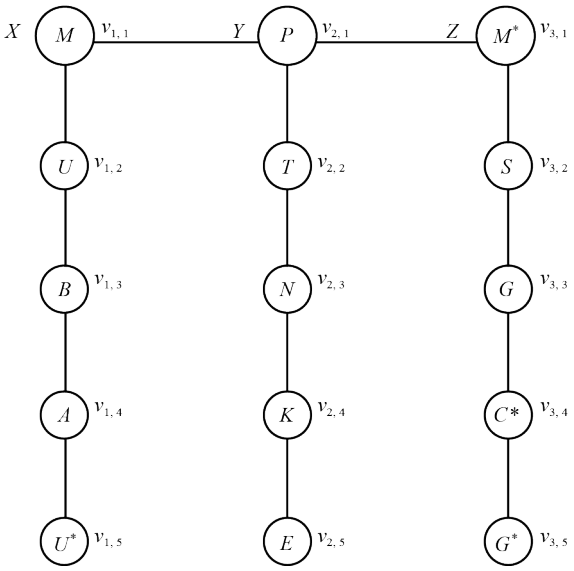

Figure 14. Comb product $P_3 \triangleright I_5$

Step 4: Calculate eccentricity centrality for each vertex of $P_3 \triangleright I_5$.

We calculate the eccentricity centrality of each node of the comb product graph by the theorem 5 presented in our paper.

Step 5: Identification of the optimal central warehouse.

The node with the maximum eccentricity centrality (i.e., with the lowest eccentricity) will be considered as the most central node in terms of maximum reachability. This node will have the shortest maximum distance to all other nodes in

the comb product graph, making it the best option for minimizing distribution time to any warehouse in the network. In $P_3 \triangleright I_5$, the vertex $v_{2,1}$ has the minimum eccentricity (i.e., maximum eccentricity centrality), which is 5. So, we can select the warehouse at Mecheda or Panskura as the central warehouse for the above-mentioned company.

Step 6: Time complexity For the comb product graph $P_3 \triangleright I_5$, $n = 3$ and $m = 5$. So, interval graph I_5 has five vertices, and Path P_3 has three vertices and comb product graph has fifteen vertices. Now, to find the eccentricities of all vertices I_H, our Algorithm ECCEN-INT needs $O(m)$ time, where $m = 5$. Again, we find the eccentricity centrality of all vertices of $P_3 \triangleright I_5$ with the help of the formula $E_C(v) = \dfrac{1}{max_{y \in V} d(v, y)} = \dfrac{1}{e(u) + q + h_2}$, where $v \in V(P_3 \triangleright I_5)$. So, to calculate the values of eccentricity centrality of all the vertices of $P_3 \triangleright I_5$, we need $O(3m)$ time, where $m = 5$. So, the overall time complexity to identify the optimal warehouse in the warehouse network we considered is $O(m) + O(3m) \approx O(m)$ time.

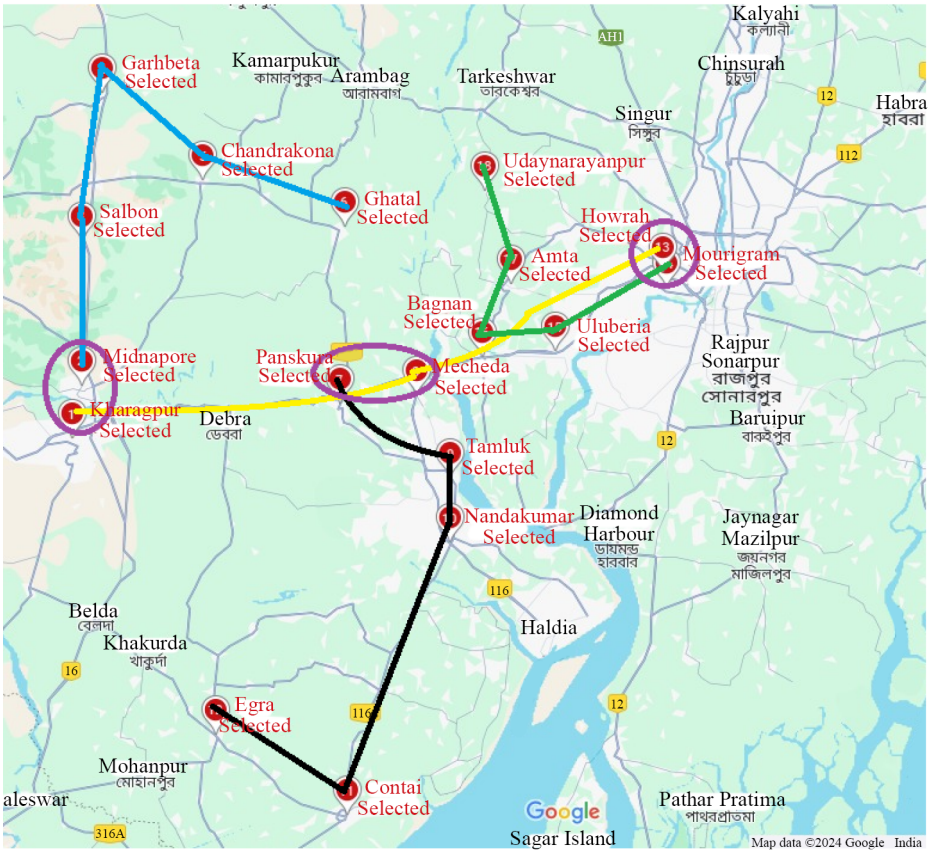

Figure 15. Real map of the comb product $P_3 \triangleright I_5$

Limitations: The comb product graph model has some limitations. The model assumes that every warehouse in the network has the exact same internal layout structure. All warehouses connect to the broader network in the same standardized way. It does not account for critical logistical constraints like delivery time windows, traffic restrictions, or variable unloading conditions. The comb product lacks flexibility for modeling dynamic changes, such as the addition of new warehouses or routes.

Note 2: In the above illustration, we did not present sensitivity analysis of our results to variations in the graph structure (e.g., when changing the number of vertices in P_n or I_m). If the number of vertices in P_n or I_m either increase or decrease then it do not affect the performance of the proposed algorithm and optimization (means select the most centrally located warehouse that minimizes the longest path to other warehouses in this network) of our solution of as our proposed

algorithm runs in $O(m)$ time (here m indicate the number of vertices of interval graph). When the value of m is large (or small), a time complexity of $O(m)$ indicates that the running time of the algorithm increases (or decreases) linearly with the size of the input. In other words, as the number of input elements doubles, the time required to complete the algorithm also approximately doubles. This means the algorithm performs one major operation for each input element. Besides these, in variations in the graph structure, the value of eccentricity of vertices may change, but in every case, we can identify the most centrally located warehouse that minimizes the longest path to other warehouses in this network.

4. Conclusion

In the present article, we have proposed some theorems related to the eccentricity centrality of the comb product between well-known graphs (K_n, S_n, W_n, P_n, and C_n) and IG. To find a BFS tree of IG, we have designed an algorithm that runs in $O(n)$ time. After that, we have studied some new results related to the diameter and eccentricity of the IG. Based on these results, we have also designed another $O(n)$ time algorithm to find the eccentricity of all vertices of interval graphs using the BFS tree. We have also developed a formula for the eccentricity centrality of the comb product between two interval graphs. We also calculate the execution time of the proposed algorithms. Besides these, we have presented a real-life application of our studied result. In the future, we shall try to develop our proposed algorithms for weighted graphs and fuzzy weighted graphs to solve more complex, realistic problems. Also, we have a plan to include a case study using real-world data in our future research to strengthen the practical applicability of the proposed model.

Data availability

In this study, we have done theoretical research work. We have only used an image from https://www.google.com/maps. No other empirical data were used or generated in this study.

Author contributions

S. Nandi and S. C. Barman conceived of the presented idea. S. Nandi and S. Mondal developed the theory and performed the computations. S. Samanta, L. Mrsic, and T. Allahviranloo verified the studied results and methodology. A. Kalampakas encouraged S. Nandi to investigate the real applications of the studied results and supervised the findings of this work. S. Nandi and S. C. Barman wrote the manuscript in consultation with T. Allahviranloo, S. Samanta, L. Mrsic, and A. Kalampakas. All authors discussed the results and contributed to the final manuscript.

Acknowledgements

We would like to express our deep and sincere gratitude to our colleagues and friends for encouraging us to complete this research work and giving continuous support throughout this research.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Trindade PC, Dreveton M, Figueiredo DR. A framework for efficient estimation of closeness centrality and eccentricity in large networks. In: *International Conference on Complex Networks*. Cham: Springer; 2025. p.13-25.

[2] Das K, Samanta S, Pal M. Study on centrality measures in social networks: a survey. *Social Network Analysis and Mining*. 2018; 8: 13. Available from: https://doi.org/10.1007/s13278-018-0493-2.

[3] Benzer S. On the topology of the genetic fne structure. *Proceedings of the National Academy of Science*. 1959; 45(11): 1607-1620. Available from: https://doi.org/10.1073/pnas.45.11.1607.

[4] Potapov AM. Multifunctionality of belowground food webs: resource, size and spatial energy channels. *Biological Reviews*. 2022; 97(4): 1691-1711. Available from: https://doi.org/10.1111/brv.12857.

[5] Huang B, Zhou M, Lu XS, Abusorrah A. Scheduling of resource allocation systems with timed Petri nets: A survey. *ACM Computing Surveys*. 2023; 55(11): 1-27. Available from: https://doi.org/10.1145/3570326.

[6] Liotta G, Rutter I, Tappini A. Parameterized complexity of graph planarity with restricted cyclic orders. In: *International Workshop on Graph-Theoretic Concepts in Computer Science*. Cham: Springer; 2022. p.383-397.

[7] Dusart J, Habib M, Corneil DG. Maximal cliques lattices structures for cocomparability graphs with algorithmic applications. *Order*. 2024; 41(1): 99-133. Available from: https://doi.org/10.1007/s11083-023-09641-x.

[8] Cao Y. Recognizing (unit) interval graphs by zigzag graph searches. In: *Symposium on Simplicity in Algorithms (SOSA)*. Society for Industrial and Applied Mathematics; 2021. p.92-106.

[9] Bairamkulov R, Friedman E. Graph fundamentals. In: *Graphs in VLSI*. Cham: Springer International Publishing; 2022. p.13-57.

[10] Kostić S, Tošković O. The time, the path, its length and strenuousness in maze learning. *Psychology*. 2022; 55(3): 313-328. Available from: https://doi.org/10.2298/PSI210301005K.

[11] Tarjan R. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*. 1972; 1(2): 146-160. Available from: https://doi.org/10.1137/0201010.

[12] Hao L, Wang Y, Bai Y, Zhou Q. Energy management strategy on a parallel mild hybrid electric vehicle based on breadth first search algorithm. *Energy Conversion and Management*. 2021; 243: 114408. Available from: https://doi.org/10.1016/j.enconman.2021.114408.

[13] Chand PK, Kumar M, Molla AR. Agent-driven bfs tree in anonymous graphs with applications. In: *International Conference on Networked Systems*. Cham: Springer; 2024. p.67-82.

[14] Saha A. Computation of average distance, radius and center of a circular-arc graph in parallel. *Journal of Physical Sciences*. 2016; 10: 171-178.

[15] Nandi S, Mondal S, Barman S. Computation of diameter, radius and center of permutation graphs. *Discrete Mathematics, Algorithms and Applications*. 2022; 14(8): 1-23. Available from: https://doi.org/10.1142/S1793830922500392.

[16] Trindade PC, Dreveton M, Figueiredo DR. A framework for efficient estimation of closeness centrality and eccentricity in large networks. In: *Complex Networks XVI: Proceedings of the 16th Conference on Complex Networks*. Cham: Springer; 2025. p.13-25.

[17] Berahmand K, Nasiri E, Forouzandeh S, Li Y. A preference random walk algorithm for link prediction through mutual influence nodes in complex networks. *Journal of King Saud University-Computer and Information Sciences*. 2022; 34(8): 5375-5387. Available from: https://doi.org/10.1016/j.jksuci.2021.05.006.

[18] Olariu S. A simple linear-time algorithm for computing the center of an interval graph. *International Journal of Computer Mathematics*. 1990; 34(3-4): 121-128. Available from: https://doi.org/10.1080/00207169008803870.

[19] Frank WT, Walter AK. Computing the eccentricity distribution of large graphs. *Algorithms*. 2013; 6(1): 100-118. Available from: https://doi.org/10.3390/a6010100.

[20] Fathalikhani K, Faramarzi H, Yousefi-Azari H. Total eccentricity of some graph operations. *Electronic Notes in Discrete Mathematics*. 2014; 45: 125-131. Available from: https://doi.org/10.1016/j.endm.2013.11.025.

[21] Puthuessery A, Kureethara J. On the centrality measures of eccentricity conected graphs. *Journal of Nonlinear Science*. 2017; 57: 1-9.

[22] Bentert M, Nichterlein A. Parameterized complexity of diameter. *Algorithmica*. 2022; 85: 325-351. Available from: https://doi.org/10.1007/s00453-022-01032-9.

[23] Kalampakas A. Fuzzy graph hyperoperations and path-based algebraic structures. *Mathematics*. 2025; 13(13): 2180. Available from: https://doi.org/10.3390/math13132180.

[24] Kwedlo W, Łubowicz M. Accelerated *K*-means algorithms for low-dimensional data on parallel shared-memory systems. *IEEE Access*. 2021; 9: 74286-74301. Available from: https://doi.org/10.1109/ACCESS.2021.3080821.

[25] Li W, Qiao M, Qin L, Zhang Y, Chang L, Lin X. Eccentricities on small-world networks. *The VLDB Journal-The International Journal on Very Large Data Bases*. 2019; 28(5): 765-792. Available from: https://doi.org/10.1007/s00778-019-00566-9.

[26] Turacı T, Durgut R. On eccentricity-based topological indices of line and para-line graphs of some convex polytopes. *Journal of Information and Optimization Sciences*. 2023; 44(7): 1303-1326. Available from: https://doi.org/10.47974/JIOS-1217.

[27] Ducoffe G. The power of interval models for computing graph centralities. *Romanian Journal of Information Technology and Automatic Control*. 2022; 32(4): 33-44.

[28] Li W, Qiao M, Qin L, Chang L, Zhang Y, Lin X, et al. On scalable computation of graph eccentricities. In: *Proceedings of the 2022 International Conference on Management of Data*. USA: Association for Computing Machinery; 2022. p.904-916.

[29] Gómez R, Gutiérrez J. Path eccentricity of graphs. *Discrete Applied Mathematics*. 2023; 337: 1-13. Available from: https://doi.org/10.1016/j.dam.2023.04.012.

[30] Lu Z, Zhou X, Zehmakan A, Zhang Z. Resistance eccentricity in graphs: distribution, computation and optimization. In: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. Utrecht, Netherlands: IEEE; 2024. p.4113-4126.