Analytical and Structural Behaviour of Centrality Measures in Tensor, Corona and Cartesian Graph Products

SUMATHI M., MANIMEKALAI S.
Department of Mathematics,
Affiliated to Bharathiar University
Dr.N.G.P Arts and science college
Kalapatti Road, Coimbatore – 641048, Tamil Nadu
INDIA

Abstract: - Graph product operations provide a way to construct complex networks from basic graphs while preserving or modifying selected structural properties. Such constructions can be used to represent systems such as transportation, communication, and distributed computing networks, where the position and importance of vertices depend on the underlying graph structure. In this study, the Tensor, Corona, and Cartesian products obtained from Complete and Wheel graphs are examined. Degree, Closeness, Betweenness, and Eigenvector centralities are considered to study how these graph products affect vertex importance, connectivity, and accessibility. Analytical expressions are developed for the structural properties of the three graph products and are examined through computational experiments using Python and NetworkX. The results show clear differences among the product constructions. In the Tensor product, the degrees of product vertices are determined by the multiplicative degrees of the factor graphs. The Corona product introduces a hierarchical structure through the attachment of Wheel graph copies to the vertices of the Complete graph. In contrast, the Cartesian product produces a comparatively more uniform distribution of structural importance. The numerical results are consistent with the analytical expressions obtained for the considered graph products. The analysis provides a comparative view of how different graph-product constructions influence vertex centrality. An additional Path–Wheel construction is also examined to provide numerical validation beyond the Complete–Wheel setting. The results provide a basis for further study of centrality behaviour in weighted, directed, and dynamic graph-product networks.

Key-Words: - Graph Products, Centrality Measures, Structural Behaviour, Tensor Product, Corona Product, Cartesian Product, Network Analysis.

Received: February 14, 2026. Revised: April 22, 2026. Accepted: June 24, 2026. Published: November 5, 2026.

1 Introduction

Graph theory studies graphs consisting of vertices and edges and provides a useful framework for representing relationships between interconnected objects. Its flexibility has led to applications in computer science, communication engineering, transportation planning, biological sciences, and social network analysis [1]. Within graph theory, graph products provide a systematic way of constructing larger graphs from simpler factor graphs while retaining selected structural properties of the original graphs [2].

Different graph product operations produce different connectivity patterns and structural arrangements. The Corona product, introduced by Frucht and Harary, constructs a hierarchical graph by attaching copies of one graph to the vertices of another graph [3]. Tensor and Cartesian products provide alternative mechanisms for combining two graphs and consequently produce different structural characteristics [2]. Such graph constructions are relevant to the study of communication systems, transportation networks, distributed computing, and infrastructure networks, where the underlying topology influences network behaviour [4].

The structural properties of graph products can also be investigated using spectral graph theory, which relates graph structure to eigenvalues and other graph characteristics [5]. Previous studies have shown that graph-product operations can influence eigenvalue distributions and centrality characteristics, leading to changes in vertex importance and connectivity patterns in the resulting graphs [6]. Studies involving Corona graph products have also examined their structural

properties resulting from the attachment of graph copies to the vertices of the base graph [7].

The study of complex networks has further increased the use of graph-theoretic methods for examining connectivity, organization, and interactions among network elements [8]. Centrality measures provide a way to compare the structural roles of vertices within a graph. Degree, Closeness, Betweenness, and Eigenvector centralities offer different perspectives on vertex connectivity, accessibility, shortest-path participation, and relationships with other important vertices [9], [11], [12]. These measures have been applied in different network settings to identify important vertices and to study structural characteristics of complex networks [13], [14], [15], [16], [17].

Recent studies have also considered the effects of graph operations and network structure on centrality distributions and related structural properties [18], [19], [20]. However, a combined analytical and computational comparison of multiple graph products using normalized centrality measures remains limited. This motivates the present study, which investigates the structural behaviour of Tensor, Corona, and Cartesian graph products constructed from Complete graphs and Wheel graphs.

In this study, analytical expressions are derived for the structural characteristics of the considered graph products, with particular emphasis on Degree centrality. The resulting graphs are further examined using normalized Degree, Closeness, Betweenness, and Eigenvector centralities. Computational experiments are performed using Python and NetworkX to compare the numerical centrality values with the corresponding analytical results. An additional Path–Wheel graph combination is also considered to provide numerical validation beyond the Complete–Wheel setting.

The main contributions of this study are the derivation of analytical expressions for the Degree centrality characteristics of Tensor, Corona, and Cartesian graph products, the comparative examination of four normalized centrality measures within a common framework, the evaluation of the analytical results through computational experiments using NetworkX, and the additional numerical validation using a Path–Wheel graph combination from a different graph family.

The present study is related to our previous work [21], which investigated Degree, Closeness, Betweenness, and Eigenvector centralities for identifying key nodes in Candida glabrata protein-protein interaction networks. That study focused on a biological network application and the computational identification of important proteins. In contrast, the present work examines the mathematical and structural behaviour of Tensor, Corona, and Cartesian graph products constructed from Complete and Wheel graphs. In particular, analytical Degree expressions are derived for these graph products, followed by computational evaluation of normalized centrality measures. Thus, although both studies consider the same four centrality measures, they differ in their graph structures, objectives, and analytical focus.

2 Literature Review

Centrality analysis is commonly used to distinguish the structural roles of vertices in a network. Different centrality measures describe vertex position in different ways. Degree Centrality is associated with the number of direct connections, while Closeness Centrality is related to the distances between a vertex and the remaining vertices. Betweenness Centrality, in turn, considers the extent to which a vertex occurs on shortest paths connecting other pairs of vertices [9]. Eigenvector Centrality uses the importance of neighbouring vertices as an additional factor when assessing a vertex [10]. Thus, the four measures considered in this study provide different descriptions of vertex importance and can be examined together to obtain a broader view of network structure [11], [12].

Several studies have applied centrality measures to biological, technological, communication, and social networks. The distribution of centrality values can provide information about network organization, connectivity, and the relative structural roles of vertices [13]. The identification of important vertices has received considerable attention in complex network analysis, particularly when the structural position of a vertex is related to the organization of connections within the network [14], [15]. Studies of clustering and community structures have also examined the relationship between local arrangements of vertices and the organization of complex networks [16]. Communicability-based investigations have further considered the relationship between graph topology and the transfer of information between vertices in complex systems [17].

Graph product operations provide a mathematical method for constructing new graphs

from two factor graphs. The resulting graph depends on the product operation as well as on the structural properties of the factor graphs. Analytical studies have examined how graph operations modify structural characteristics and connectivity patterns [18]. Comparative investigations have also considered the influence of different graph construction mechanisms on centrality distributions and the structural roles of vertices [19]. Graph products have additionally been investigated as mathematical models for interconnected systems, including communication and infrastructure networks, because different product operations produce different forms of connectivity [20].

Although previous studies have examined individual graph products, graph invariants, and different centrality measures, comparatively fewer studies consider several graph-product operations and multiple normalized centrality measures within the same analytical and computational framework. In particular, the combined investigation of Tensor, Corona, and Cartesian graph products using Degree, Closeness, Betweenness, and Eigenvector centralities remains limited. Therefore, the present study examines these three graph products constructed from Complete and Wheel graphs using analytical expressions and NetworkX-based computational analysis. The study compares the centrality characteristics produced by the different product operations and examines how their structural differences are reflected in the resulting vertex centralities.

3 Preliminaries

Let G = (V(G), E(G)) and H = (V(H), E(H)) be two simple connected graphs, where V(G) and V(H) denote the vertex sets and E(G) and E(H) denote the edge sets, respectively. In this section, we recall the definitions and fundamental properties of the base graphs and graph product operations considered in this study.

3.1 Complete Graph

A complete graph K_n is a simple graph in which every pair of distinct vertices is connected by an edge. Formally, for all $u,v \in V(K_n)$ with $u \neq v$, we have $uv \in E(K_n)$. Each vertex in K_n has degree deg(v) = n-1 and the total number of edges is $|E(K_n)| = \frac{n(n-1)}{2}$. Figure 1 illustrates the structure of the complete graph K_n considered in this study.

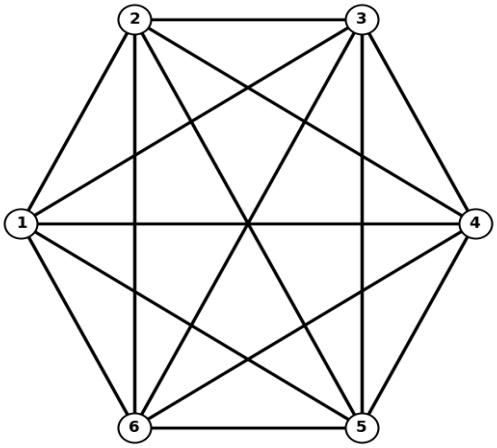


Fig. 1: Complete Graph
Source: created by the authors

3.2 Wheel Graph

A wheel graph W_m is obtained by joining a single hub vertex to all vertices of a cycle C_{m-1}. Thus $W_m = K_1 + C_{m-1}$. In W_m , the hub vertex has degree $m-1$, while each rim vertex has degree 3. The wheel graph combines cyclic symmetry with centralized connectivity. The total number of edges in W_m is $|E(W)| = 2(m-1)$. Figure 2 illustrates the structure of the wheel graph W_m, showing its central hub and rim vertices.

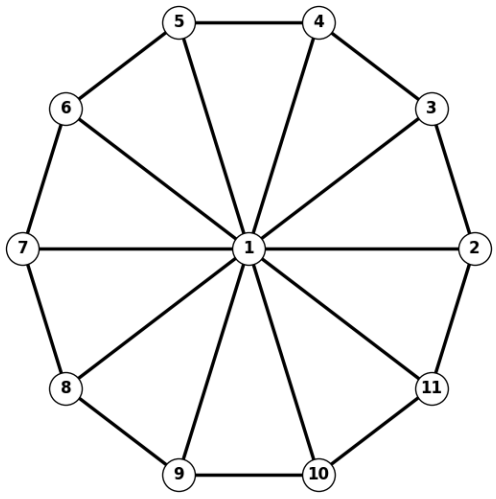


Fig. 2: Wheel Graph
Source: created by the authors

3.3 Path Graph

A Path graph, denoted by P_n, is a simple connected graph in which the vertices are arranged in a linear sequence. It consists of n vertices and $n-1$ edges. The two end vertices have degree one, while every internal vertex has degree two. Path graphs are

widely used to model sequential communication, transportation routes, and linear network structures. Figure 3 illustrates the Path graph used for the additional numerical validation.

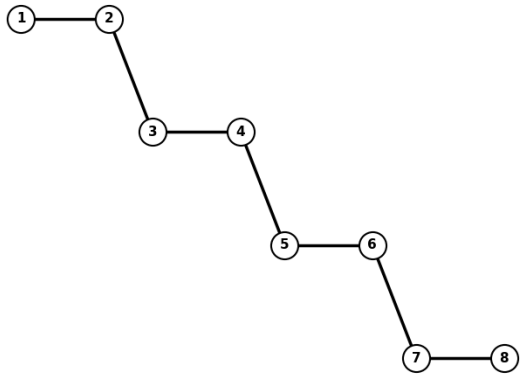


Fig. 3: Path Graph
Source: created by the authors

3.4 Tensor Product

The Tensor product of graphs G and H, denoted by G×H, is defined on the vertex set V(G×H) = V(G) ×V(H), where two vertices (g_1,h_1) and (g_2, h_2) are adjacent in G×H if and only if $g_1\ g_2 \in$ E(G) and $h_1h_2 \in$E(H). The resulting graph contains |V(G)|·|V(H)| vertices.

3.5 Cartesian Product

The Cartesian product G□H , denoted by G□H, is defined on the vertex set V(G □ H) = V(G)×V(H). Two vertices (g_1, h_1) and $(g_2, h_2)$are adjacent if either $g_1 = g_2$ and $h_1h_2 \in$E(H), or $h_1 = h_2$ and $g_1g_2 \in$E(G). The Cartesian product preserves additive connectivity between factor graphs.

3.6 Corona Product

The Corona product G⊙H is obtained by taking one copy of G and |V(G)| copies of H, and joining the i^{th} vertex of G to every vertex in the i^{th} copy of H. Consequently, the total number of vertices in G⊙H is |V(G⊙H)|=|V(G)|+|V(G)|·|V(H)|. This construction introduces a hierarchical attachment mechanism centered around the vertices of G.

4 Theoretical Results

4.1 Theorem 1 (Tensor Product – Degree Amplification)

Let G = K_n and H = W_m. In the Tensor Product $K_n \times W_m$, the vertices corresponding to the hub of W_m achieve maximum degree centrality.

Proof

For the Tensor product:$\deg(u,v)=\deg_G(u)\cdot\deg_H(v)$. Since every vertex of K_n has degree n-1, $\deg_{K_n}(u)$ = n-1 and W_m ,For Wheel graph: Hub degree = m–1, Rim degree = 3.Thus Hub-intersection vertex degree:$(n-1)(m-1)$. Rim-intersection vertex degree 3(n-1) for m>4, $(n-1)(m-1)>3(n-1)$. Therefore hub-intersection vertices have strictly greater degree centrality.

4.2 Theorem 2 (Corona Product – Hierarchical Betweenness Dominance $K_n \odot W_m$)

Let G= K_n and H =W_m. In the Corona Product $K_n \odot W_m$, every base vertex of K_n has strictly greater betweenness centrality than any vertex belonging to the attached copies of W_m.

Proof

By definition of the Corona product $K_n \odot W_m$, each vertex of K_n is connected to all vertices of its corresponding copy of $(W_m^{(i)})$. Consider a base vertex v_i. For every j ≠ i, every shortest path between a vertex in $(W_m^{(i)})$ and a vertex in $W_m^{(j)}$ is of the form u-v_i-v_j-w, where $u \in V(W_m^{(i)})$ and $w \in W_m^{(j)}$There are m^2 such vertex pairs for each $j \neq i$. Hence, these pairs contribute (n-1)m^2shortest paths through v_i . Similarly, for every $j \neq i$, each vertex of $W_m^{(i)}$and the base vertex vj has the unique shortest path u-v_i-v_j. Therefore, these pairs contribute (n-1)m additional shortest paths through v_i. Consequently, the number of shortest-path pairs for which v_i is an internal vertex is at least (n-1)(m^2+m). Now consider any vertex u∈ $W_m^{(i)}$. Since every vertex of $W_m^{(i)}$ is directly connected to the base vertex v_i, no shortest path connecting a vertex of $W_m^{(i)}$ to a vertex outside that copy requires u as an internal vertex. Thus, u can contribute only to shortest paths whose two endpoints lie within the same copy $W_m^{(i)}$. Since u itself cannot be an endpoint of a path for which it is an internal vertex, the two endpoints must be selected from the remaining m−1 vertices. Hence, the number of possible endpoint pairs is at most$\binom{m-1}{2}$. Each such pair contributes at most 1 to the betweenness score. Therefore, BC(u)≤ $\binom{m-1}{2}$. For n≥2 and m≥4,(n-1)$(m^2+m)>\frac{(m-1)(m-2}{2}$. Hence, BC($v_i$)>BC($u$). Since the same normalization factor is applied to all

vertices in the normalized betweenness centrality, the ordering is preserved. Therefore, BC(v_i)>BC(u) for all $v_i \in V(K_n)$ and $u \in V(W_m^{(i)})$. ■

4.3 Theorem 3 (Cartesian Product – Structural Balance)

Let G=K_n and H=W_m. In the Cartesian Product $K_n \square W_m$. the degree of a vertex (u,v) is given deg $(u,v)=\deg_{K_n}(u)+\deg_{W_m}(v)$. Since every vertex of K_n has degree $n-1$, and the hub and rim vertices of W_m have degrees $m-1$ and 3, respectively, the maximum degree occurs at vertices corresponding to the hub of W_m. Therefore, the degree of a hub-intersection vertex is $(n-1)+(m-1) = n+m-2$, while the degree of a rim-intersection vertex is $(n-1)+3 = n+2$.

Since the Cartesian product combines degrees additively rather than multiplicatively, the exact difference between the maximum and minimum degrees (degree spread) is given by $(n+m-2)-(n+2) = m-4$. In contrast, the corresponding degree spread in the Tensor product$K_n \times W_m$ is $(n-1)(m-1)-3(n-1) = (n-1)(m-4)$.Thus, for $n>1$ and m>4, the degree spread of the Tensor product is exactly $(n-1)$ times that of the Cartesian product. For $n>2$, the Tensor product therefore has a strictly larger degree spread, while for $n=2$, the two degree spreads are equal. This demonstrates that the Cartesian product produces a more balanced degree distribution than the Tensor product when $n>2$. ■

4.4 General Degree Formulas for K_n and W_m

The degree values for the three graph products considered in this work are listed in Table 1 For the Tensor product $K_n \times W_m$, consider a vertex (u,v), where (u) is a vertex of K_n and (v) is a vertex of W_m. Its degree is given by $\deg(u,v)= \deg_{K_n}(u)+ \deg_{W_m}(v)$. Here, $\deg_{K_n}(u)$ denotes the degree of the vertex u in K_n, $\deg_{W_m}(v)$ denotes the degree of the vertex v in W_m. Since every vertex of K_n has degree $n-1$, a vertex corresponding to the hub of W_m, whose degree is $m-1$, has degree (n-1)(m-1). Similarly, each rim vertex of W_m has degree 3, and hence the corresponding vertex in the Tensor product has degree $3(n-1)$. Therefore, the vertices corresponding to the hub of W_m have the maximum degree in the Tensor product.

Table 1. Degree Formulas for Tensor, Corona and Cartesian Products of K_n and W_m

Product Type	Vertex Class	Degree Formula
Tensor Product $K_n \times W_m$	Hub Vertex	$(n-1)(m-1)$
Tensor Product $K_n \times W_m$	Rim Vertex	$3(n-1)$
Cartesian Product $K_n \square W_m$	Hub Vertex	$(n+m-2)$
Cartesian Product $K_n \square W_m$	Rim Vertex	$(n+2)$
Corona Product $K_n \odot W_m$	Base Vertex	$(n+m-1)$
Corona Product $K_n \odot W_m$	Attached Rim	4
Corona Product $K_n \odot W_m$	Attached Hub	(m)

Source: created by the authors

For the Cartesian product $K_n \square W_m$, consider a vertex (u,v) where $u \in V(K_n)$ and $v \in V(W_m)$. The degree of (u,v) is the sum of the degrees of (u) and (v) in their respective graphs: $\deg(u,v)= \deg_{K_n}(u)+ \deg_{W_m}(u)$. Since the degree of every vertex in K_n is $(n-1)$, and the hub and rim vertices of W_m have degrees $(m-1)$ and 3, respectively, the degree of a hub-intersection vertex is $((n-1)+(m-1) = n+m-2)$. For a rim-intersection vertex, the degree is $(n-1)+3 = n+2$ Hence, the difference between the maximum and minimum degrees is $(n+m-2)-(n+2) = m-4$

For the Corona product $K_n \odot W_m$, one copy of W_m is attached to each vertex of K_n. Each base vertex of K_n is connected to all m vertices of its corresponding copy of W_m. Since a base vertex has degree $n-1$ in K_n, its degree in the Corona product is $(n-1)+m = n+m-1$.The attached hub vertex is connected to the base vertex and to all $m-1$ rim vertices of its wheel copy. Therefore, its degree is $1+(m-1) = m$. Each attached rim vertex is connected to the hub, to its two neighbouring rim vertices, and to the corresponding base vertex. Hence, its degree is 1+2+1 = 4.

Thus, the degree expressions in Table 1 show three different degree patterns. The Tensor product gives a multiplicative degree expression, the Corona product produces separate degree values for the base and attached vertices, and the Cartesian product follows an additive degree expression.

5 Centrality Measures

Let G = (V,E) be a simple connected graph with |V| = N.

Degree Centrality: The Degree Centrality of a vertex $v \in V$ is defined as $C_D(v) = \frac{\deg(v)}{N-1}$, where $\deg(v)$ denotes the degree of vertex v.

Betweenness Centrality: The Betweenness Centrality of a vertex v is defined as $C_B(v) = \sum_{S \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where σ_{st} is the total number of shortest paths between vertices s and t, $\sigma_{st}(v)$ is the number of those shortest paths that pass through v. For an undirected graph with N vertices, the normalized betweenness centrality is obtained using the normalization factor $\frac{2}{(N-1)(N-2)}$

Closeness Centrality: The Closeness Centrality of vertex v is defined as $C_C(v) = \frac{N-1}{\sum_{u \neq v} d(u,v)}$, where $d(u,v)$ denotes the shortest-path distance between vertices v and u.

Eigenvector Centrality: The Eigenvector Centrality of vertex v is defined as $C_E(v) = \frac{1}{\lambda} \sum_{u \epsilon N(v)} A_{vu} C_E(u)$, where A is the adjacency matrix of the graph, λ is the largest eigenvalue of A. In the computational analysis, the eigenvector centrality values are normalized according to the convention implemented in NetworkX.

Table 2, Table 3 and Table 4 contain the centrality values obtained for the three graph products. For each graph, Degree, Closeness, Betweenness, and Eigenvector centralities were calculated using the normalized form. Table 2 corresponds to $(K_6 \times W_{11})$, Table 3 corresponds to $(K_6 \odot W_{11})$, and Table 4 corresponds to $(K_6 \square W_{11})$. The entries are grouped according to the vertex classes in each product graph.

Table 2. Tensor Product of $K_6 \times W_{11}$ Centrality

Node	Degree	Close Ness	Between Ness	Eigen vector
1	0.7692	0.8125	0.0954	0.2413
2 to 11	0.2308	0.5652	0.0028	0.1041
12	0.7692	0.8125	0.0954	0.2413
13 to 22	0.2308	0.5652	0.0028	0.1041
23	0.7692	0.8125	0.0954	0.2413
24 to 33	0.2308	0.5652	0.0028	0.1041
34	0.7692	0.8125	0.0954	0.2413
35 to 44	0.2308	0.5652	0.0028	0.1041
45	0.7692	0.8125	0.0954	0.2413
46 to 55	0.2308	0.5652	0.0028	0.1041
56	0.7692	0.8125	0.0954	0.2413
57 to 66	0.2308	0.5652	0.0028	0.1041

Source: created by the authors

Table 3. Corona Product of $K_6 \odot W_{11}$ Centrality

Node	Degree	Close Ness	Between Ness	Eigen vector
1 to 6	0.2254	0.5635	0.2720	0.3053
7	0.1549	0.3817	0.0064	0.1334
8 to 17	0.0563	0.3679	0.0001	0.0746
18	0.1549	0.3817	0.0064	0.1334
19 to 28	0.0563	0.3679	0.0001	0.0746
29	0.1549	0.3817	0.0064	0.1334
30 to 39	0.0563	0.3679	0.0001	0.0746
40	0.1549	0.3817	0.0064	0.1334
41 to 50	0.0563	0.3679	0.0001	0.0746
51	0.1549	0.3817	0.0064	0.1334
52 to 61	0.0563	0.3679	0.0001	0.0746
62	0.1549	0.3817	0.0064	0.1334
63 to 72	0.0563	0.3679	0.0001	0.0746

Source: created by the authors

Table 4. Cartesian Product of $K_6 \square W_{11}$ Centrality

Node	Degree	Close Ness	Between Ness	Eigen vector
1	0.2307	0.5652	0.1225	0.2412
2 to 11	0.1231	0.4140	0.0110	0.1041
12	0.2307	0.5652	0.1225	0.2412
13 to 22	0.1231	0.4140	0.0110	0.1041
23	0.2307	0.5652	0.1225	0.2412
24 to 33	0.1231	0.4140	0.0110	0.1041
34	0.2307	0.5652	0.1225	0.2412
35 to 44	0.1231	0.4140	0.0110	0.1041
45	0.2307	0.5652	0.1225	0.2412
46 to 55	0.1231	0.4140	0.0110	0.1041
56	0.2307	0.5652	0.1225	0.2412
57 to 66	0.1231	0.4140	0.0110	0.1041

Source: created by the authors

5.1 Structural Behaviour of the Tensor Product

In the Tensor product $K_n \times W_m$, adjacency depends on both factor graphs. Since every vertex of K_n has degree n-1, the degree of a product vertex changes according to whether its corresponding vertex in W_m is the hub or a rim vertex. For a hub-intersection vertex, the degree is (n-1)(m-1), whereas for a rim-intersection vertex it is 3(n-1) for m>4, (n-1)(m-1)>3(n-1). Thus, the hub-intersection vertices have the larger degree in the Tensor product. This is also the result obtained in Theorem 1.

For $K_6 \times W_{11}$, the hub-intersection vertices have normalized Degree centrality **0.7692**, while the rim-intersection vertices have **0.2308**. The Closeness centrality values are **0.8125** and **0.5652**, respectively. For Betweenness centrality, the corresponding values are **0.0954** and **0.0028**. The Eigenvector centrality values are **0.2413** for the

hub-intersection vertices and **0.1041** for the rim-intersection vertices.

The four measures therefore give the same ordering of the two vertex classes. The hub-intersection vertices have higher values than the rim-intersection vertices in Degree, Closeness, Betweenness, and Eigenvector centralities. The difference is most noticeable in Degree and Closeness centralities.

For this $K_6 \times W_{11}$ example, the numerical values follow the degree pattern obtained from the product construction. The wheel hub has more neighbours than a rim vertex in W_{11}, and this difference is retained after forming the Tensor product. Hence, the hub-intersection vertices form the more prominent vertex class in the resulting graph

5.2 Structural Behaviour of the Corona Product

Consider the corona product $K_n \odot W_m$**.** A copy of W_m is attached to each vertex of K_n. Hence, the vertices in the complete graph and the vertices in the attached wheel copies do not have the same role in the resulting graph. Let v_i be a vertex of K_n. It has n-1 neighbours in K_n, and it is also joined to all the m vertices of its corresponding wheel copy. Therefore, deg(v_i) = (n-1)+m = n+m-1.

For a vertex in an attached wheel, the degree is smaller. The hub is adjacent to the m-1 rim vertices and to the corresponding base vertex v_i, giving degree m. Each rim vertex has two rim neighbours, the hub, and the base vertex v_i. Its degree is therefore 4. Thus, the three vertex classes in $\mathbf{K_n \odot W_m}$ have degrees n+m-1, m, and 4, respectively.

The base vertices also play an important role in shortest paths connecting different wheel copies. Consequently, their Betweenness centrality is higher than that of vertices within the attached wheel copies, in agreement with Theorem 2.

The degree and betweenness results therefore distinguish the base vertices from the attached wheel vertices. A base vertex is connected to the other vertices of $\mathbf{K_n}$ as well as to all vertices in its corresponding $\mathbf{W_m}$, while the vertices inside a wheel copy mainly account for local connections.

Hence, the corona product $\mathbf{K_n \odot W_m}$ has a connecting layer formed by the vertices of $\mathbf{K_n}$ and local structures formed by the attached wheel copies. This difference explains the higher importance of the base vertices in Betweenness centrality.

5.3 Structural Behaviour of the Cartesian Product

The Cartesian product combines two graphs while preserving the adjacency characteristics of both constituent graphs. Unlike the tensor product, which creates simultaneous adjacency, and the corona product, which introduces attachment of a wheel copy to each base vertex, the Cartesian product combines the neighbourhoods of the two factor graphs.

For $K_n \square W_m$, the degree of a product vertex is obtained from the degrees of the corresponding vertices inK_n and W_m. Thus, the hub-intersection and rim-intersection vertices have degrees n+m−2 and n+2, respectively, as established in Theorem 3.

For $K_6 \square W_{11}$, the hub-intersection vertices have Degree, Closeness, Betweenness, and Eigenvector centralities of **0.2307, 0.5652, 0.1225, and 0.2412**, respectively. The corresponding values for the rim-intersection vertices are **0.1231, 0.4140, 0.0110, and 0.1041**.

The hub-intersection vertices therefore have higher values for all four measures. The numerical results are consistent with the degree relation obtained analytically for the Cartesian product.

6 Computational Methodology

The centrality calculations were carried out for the Tensor, Corona, and Cartesian products of $\mathbf{K_n}$ and $\mathbf{W_m}$. Python (version 3.13.0) and the NetworkX (version 3.4.2) were used for the graph construction and numerical calculations. The same software settings were used for all the product graphs. For a graph with N vertices, Degree Centrality was calculated as $C_D = \frac{\deg(v)}{N-1}$. Thus, the degree of each vertex is divided by the maximum possible degree (N-1). Closeness Centrality was calculated from the shortest-path distances from the selected vertex to the other reachable vertices. Betweenness Centrality was obtained with the Brandes algorithm. Normalization was enabled in the calculation. For an undirected graph, the normalization factor used for Betweenness Centrality is $\frac{2}{(N-1)(N-2)}$.

Eigenvector Centrality was calculated using the power-iteration method available in NetworkX, with a tolerance of 10^{-6}. The values obtained from this calculation were normalized according to the NetworkX convention. The numerical Eigenvector Centrality values reported in this paper were obtained using this setting.

The graphs $\mathbf{K_n}$ and $\mathbf{W_m}$ were generated before constructing the product graphs. The three products used in the study were $K_n \times W_m$, $\mathbf{K_n \odot W_m}$, and $K_n \square W_m$. Their adjacency information was used in the centrality calculations. For K_6 and W_{11}, the Tensor product has 66 vertices and the Cartesian product also has 66 vertices. The Corona product has 72 vertices. This is because six copies of W_{11} are attached to the six vertices of K_6.

Graph labelling and maximum-value selection
The same vertex numbering was used throughout the calculations. The vertices of the factor graphs were assigned integer labels. In the product graphs, the vertices were identified using the corresponding vertex pairs. No change in the labelling scheme was made between the three product constructions.

The four centrality measures were calculated for every vertex in each product graph. The vertices were then arranged according to their structural type. For example, the relevant classes include hub and rim vertices in the Tensor and Cartesian products and base, attached-hub, and attached-rim vertices in the Corona product. The values within each class were compared. The largest value was taken as the representative maximum for that class. When several vertices gave the same maximum, the same value was reported for those vertices. The following steps were used for the numerical analysis.

Algorithm 1: Computational Procedure for Centrality Analysis
Input: K_n and W_m
Output: Normalized centrality values
1. Generate K_n and W_m.
2. Construct $K_n \times W_m$, $K_n \odot W_m$, and $K_n \square W_m$.
3. Compute Degree, Closeness, Betweenness, and Eigenvector centralities for all vertices.
4. Use the normalization settings of NetworkX
5. Divide the vertices into their respective structural classes
6. Find the maximum centrality value in each class.
7. Compare these values with the analytical results.
8. Repeat the calculation for different values of *n* and *m*.

6.1 Algorithmic Complexity Analysis
The time required for the calculations depends on the centrality measure and the graph representation. For an adjacency-list representation, the degree of all vertices can be obtained in O(|V|+|E|) time.
For the unweighted graphs used here, Closeness Centrality requires shortest-path calculations and has a time complexity of O(|V||E|). Betweenness Centrality was calculated using the Brandes algorithm, which also requires O(|V||E|) time for unweighted graphs.

The Eigenvector Centrality calculation uses power iteration. If k iterations are needed to satisfy the convergence condition, the approximate computational cost is O(k|E|). Here, k is the number of iterations required for convergence. These complexity values correspond to the algorithms used in the present NetworkX-based implementation.

7 Numerical Validation
To examine the analytical results for more than one choice of *n* and *m*, numerical tests were carried out for the Tensor, Corona, and Cartesian products of the Complete graph K_n and Wheel graph W_m. Degree, Closeness, Betweenness, and Eigenvector centralities were calculated for the vertices of each product graph using the procedure given in Section 6. The obtained values were then compared with the results derived in Section 4.

Table 5 gives the maximum normalized centrality values for the three graph products. Different values can be seen for Degree, Eigenvector, Closeness, and Betweenness centralities. The results follow the structural differences of the three product operations and agree with the degree relations obtained in Section 4. The corresponding centrality patterns are discussed in the next section.

Table 5. Maximum Normalized Centrality Measures of the Tensor, Corona, and Cartesian Products of K_6 and W_{11}

Graph Product	Degree	Eigen vector	Close Ness	Between ess
Tensor	0.7692	0.2413	0.8125	0.0954
Corona	0.2254	0.3053	0.5635	0.2720
Cartesian	0.2307	0.2412	0.5652	0.1225

Source: created by the authors

The highest normalized Degree, Closeness, Betweenness, and Eigenvector values for P_8 and W_{11} are given in Table 6 for the three graph products.

Table 6. Maximum Normalized Centrality Measures of the Tensor, Corona, and Cartesian Products of P_8 and W_{11}

Graph Product	Degree	Eigen vector	Close Ness	Between ness
Corona	0.1368	0.2650	0.3393	**0.5975**
Tensor	**0.2299**	**0.2744**	**0.4350**	0.1909
Cartesian	0.1379	0.2743	0.3398	0.2208

Source: created by the authors

For further numerical comparison, the Path graph P_8 and the Wheel graph W_{11} were used. The maximum normalized centrality values are listed in Table 6. Figure 4 gives the comparison of the maximum centrality values for the Tensor, Corona, and Cartesian products of K_6 and W_{11}.

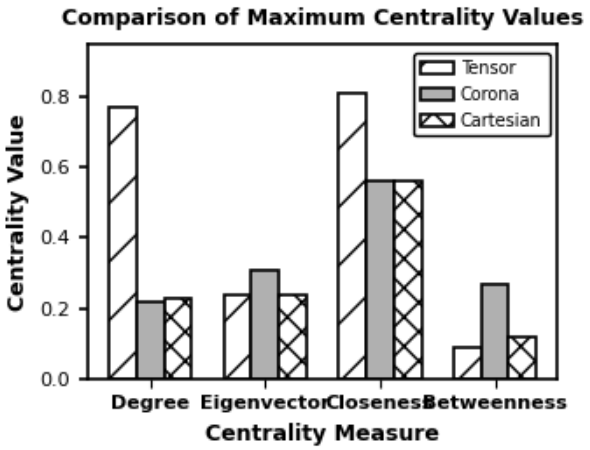


Fig. 4: Comparison of maximum normalized centrality values for the Tensor, Corona, and Cartesian products of K_6 and W_{11}
Source: created by the authors

Figure 4 shows the variation in the four centrality measures for the three graph products. The values shown in the figure correspond to the numerical results in Table 5. They also follow the degree relations obtained in Section 4.

8 Discussion

8.1 Structural Interpretation of Centrality Behaviour

The Path graph is also used in the numerical study. The values obtained for this graph give a further comparison of the three products. The Tensor, Corona, and Cartesian products are constructed using the same factor graphs. However, the way in which the vertices are joined is different for each product. This difference is reflected in the Degree, Closeness, Betweenness, and Eigenvector centrality values.

The results in Sections 4 and 7 help to explain these values. For the Tensor product, the degree of a vertex is obtained from the degrees of the two corresponding vertices. In the Corona product, each vertex of the base graph is joined to all vertices in its associated copy of the wheel graph. For the Cartesian product, the degrees from the two factor graphs are added. Thus, the three products give different degree and connectivity patterns.

The numerical values obtained in Section 7 follow the degree expressions given in Section 4. The centrality values therefore change with the structure of the product graph. The three products do not give the same distribution of vertex importance.

Table 7 (Appendix) gives a qualitative summary of the three graph products. It lists the connectivity pattern, dominant vertices, degree distribution, structural behaviour, network characteristics, and possible applications. Section 7 gives the numerical results, whereas Table 7 (Appendix) gives a qualitative description of those results.

8.2 Structural Indicators and Network Resilience

The centrality values indicate how vertex importance is distributed in the three graph products. In the Tensor product, the hub-intersection vertices have higher centrality values. In the Corona product, the base vertices have greater influence since they connect the attached copies. The Cartesian product shows a more even distribution of structural importance.

These results give structural information about the product graphs. They are not direct measures of network resilience under vertex failure. Failure-based analysis and detailed connectivity tests are not considered in the present work.

8.3 Practical Implication

The analytical and computational results give different structural patterns for the three graph products. The Tensor product has multiplicative connectivity with greater concentration around the hub-intersection vertices. The Corona product has a hierarchical arrangement between the base graph and the attached copies. The Cartesian product has a more balanced connectivity pattern.

These differences can be useful when choosing a graph product for mathematical modelling of

communication, hierarchical, transportation, and interconnection networks.

9 Conclusion

This work studied the structural behaviour of Tensor, Corona, and Cartesian graph products formed from the Complete graph $\mathbf{K_n}$ and the wheel graph $\mathbf{W_m}$. Degree, Closeness, Betweenness, and Eigenvector centralities were used to compare the three products and to study the distribution of vertex importance in the resulting graphs.

Degree expressions were obtained for the different vertex classes in each product. These expressions were then checked using numerical calculations. The numerical values obtained from the computational study agree with the analytical results and show the effect of the product operation on the centrality values.

The three products show different structural patterns. In the Tensor product, the degree grows multiplicatively and the hub-intersection vertices have greater structural importance. The Corona product has a hierarchical arrangement, with the base vertices connecting the attached copies. In the Cartesian product, the degree contributions are added and the distribution of structural importance is more balanced.

The results can be useful in selecting a suitable graph product for modelling different types of network structures, including communication, transportation, distributed computing, and interconnection networks. The choice of product can therefore be made according to the connectivity pattern required in the model.

The additional experiment using the Path graph P_8 and the Wheel graph W_{11} gives another numerical comparison and supports the observations obtained from the Complete graph case.

In future work, the analysis can be extended to weighted, directed, dynamic, and hypergraph-based products. Other graph invariants and centrality measures can also be included to obtain a broader comparison of these graph constructions.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

[1] Bondy, J. A., & Murty, U. S. R. (2008). Graph Theory. Springer.

[2] Imrich, W., & Klavžar, S. (2000). Product Graphs: Structure and Recognition. Wiley.

[3] Frucht, R., & Harary, F. (1970). On the corona of two graphs. Aequationes Mathematicae, 4(3), 322–325.

[4] Newman, M. E. J. (2010). Networks: An Introduction. Oxford University Press.

[5] Cvetković, D. M., Doob, M., & Sachs, H. (1995). Spectra of Graphs. Johann Ambrosius Barth.

[6] Ilić, A. (2010). Distance of eigenvalues and centrality measures in product graphs. Applied Mathematics Letters, 23(10), 1261–1266.

[7] Khalifeh, M. H., Azari, H., & Dandighan, A. (2009). Wiener, Szeged, and PI indices of corona product of graphs. Discrete Applied Mathematics, 157(1), 1–10.

[8] Estrada, E. (2012). The Structure of Complex Networks: Theory and Applications. Oxford University Press.

[9] Freeman, L. C. (1979). Centrality in social networks: Conceptual clarification. Social Networks, 1(3), 215–239.

[10] Bonacich, P. (1987). Power and centrality: A family of measures. American Journal of Sociology, 92(5), 1170–1182.

[11] Borgatti, S. P., & Everett, M. G. (2006). A graph-theoretic perspective on centrality. Social Networks, 28(4), 466–484.

[12] Barabási, A.-L. (2016). Network Science. Cambridge University Press.

[13] Li, M., Lu, Y., & Wu, J. (2018). Centrality measures in complex networks: A survey. Physica A, 490, 152–167.

[14] da Silva, F. N., & Costa, L. da F. (2019). Characterizing complex networks via centrality distributions. Chaos, 29(6), 063125.

[15] Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C., & Zhou, T. (2016). Vital nodes identification in complex networks. Physics Reports, 650, 1–63.

[16] Malliaros, F. D., & Vazirgiannis, M. (2013). Clustering and community detection in directed networks. Physics Reports, 533(4), 95–142.

[17] Estrada, E., & Hatano, N. (2008). Communicability in complex networks. Physical Review E, 77(3), 036111.
[18] Ahmad, I., & Bhat, M. I. (2020). Centrality analysis of graph operations and network robustness. Journal of Applied Mathematics and Computing, 62(1–2), 289–305.
[19] Zhao, J., Wu, J., & Xu, K. (2021). Comparative analysis of centrality measures in composite networks. Applied Network Science, 6(1), 45.
[20] Kumar, S., & Singh, R. (2022). Structural influence of graph products in communication networks. IEEE Access, 10, 45321–4533.
[21] Sumathi M, Manimekalai S. Analysing centrality measures for key node identification in Candida glabrata protein networks. Indian J Microbiol Res. 2025;12(2).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US

APPENDIX

Table 7. Structural Characteristics of Tensor, Corona and Cartesian Graph Products

Structural Property	Tensor Product	Corona Product	Cartesian Product
Connectivity Pattern	Multiplicative	Hierarchical	Additive
Dominant Vertices	Hub-intersection vertices	Base vertices	Uniformly distributed vertices
Degree Distribution	Highly concentrated	Moderately concentrated	Balanced
Information Flow	Concentrated structural influence	Hierarchical organization	Comparatively balanced distribution
Structural Behaviour	Hub amplification	Routing dependency	Balanced structural distribution
Network Characteristic	Concentrated structural importance	Hierarchical organization	Balanced connectivity
Typical Applications	Communication networks	Organizational networks	Transportation and parallel computing

Source: created by the authors