J. Appl. Comput. Mech., 9(2) (2023) 384-403
DOI: 10.22055/jacm.2022.41255.3721

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran
University of Ahvaz

Journal of
Applied and Computational Mechanics

Research Paper

Automatic Structural Synthesis of Planetary Geared Mechanisms using Graph Theory

Hind A. Nafeh[1], Essam L. Esmail [2], Sajad H. Abdali[3]

[1] Department of Mechanical Engineering, University of Al-Qadisiyah, Diwaniyah, 58001, Iraq, Email: eng.mec.mas.20.3@qu.edu.iq

[2] Department of Mechanical Engineering, University of Al-Qadisiyah, Diwaniyah, 58001, Iraq, Email: essam.esmail@qu.edu.iq

[3] Department of Mechanical Engineering, University of Al-Qadisiyah, Diwaniyah, 58001, Iraq, Email: eng.mech.4st-11@qu.edu.iq

Received July 03 2022; Revised August 26 2022; Accepted for publication September 23 2022.
Corresponding author: E.L. Esmail (essam.esmail@qu.edu.iq)

Abstract. Graphs are an effective tool for planetary gear trains (PGTs) synthesis and for the enumeration of all possible PGTs for transmission systems. In the past fifty years, considerable effort has been devoted to the synthesis of PGTs. To date, however, synthesis results are inconsistent, and accurate synthesis results are difficult to achieve. This paper proposes a systematic approach for synthesizing PGTs depending on spanning trees and parent graphs. Trees suitable for constructing rooted graphs are first identified. The parent graphs are then listed. Finally, geared graphs are discovered by inspecting their parent graphs and spanning trees. To precisely detect spanning trees, a novel method based on two link assortment equations is presented. Transfer vertices and edge levels are detected without the use of any computations. This work develops the vertex matrix of the rooted graph, and its distinctive equation is used to arrange the vertex degree arrays according to the vertex levels and eliminate the arrays that violate the distinctive equations. The precise results of the 5-link geared graphs are confirmed to be 24. The disparity between the recent and previous synthesis results can be attributed to the fact that the findings of the current method, which employs rooted graphs, are more comprehensive than those obtained with graphs lacking multiple joints. A novel algorithm for detecting structural isomorphism is proposed. By comparing the vertex degree listings and gear strings, non-isomorphic geared graphs are obtained. The algorithm is simple and computationally efficient. The graph representation is one-to-one with the vertex degree listing and gear string representation. This allows for the storage of a large number of graphs on a computer for later use.

Keywords: Displacement graph, link assortment array, rooted parent graph, pseudo-isomorphism graph, structure synthesis, spanning tree.

1. Introduction

1.1 Background

Planetary gear trains (PGTs) are one of the earliest mechanical systems still in use today and are frequently found in machinery. PGTs have significant size and weight advantages over conventional gear trains. A PGT is profitable in that it can achieve a high transmission ratio and a small design. PGTs are therefore widely utilized in contemporary machinery such as tool machine gearboxes, wind turbines, hoists, robotic wrists, etc.

Planetary gear mechanisms (PGMs) are the subject of a large amount of scientific literature, which attests to their expanding industrial importance. For each unique design task in the development of PGMs, achieving a better-performing kinematic structure has been a challenging but crucial problem [1, 2]. For a very long time, two methods were employed: first, researchers relied significantly on the concept of potential structures and selected one based on their experience and intuition. Second, based on the experience of researchers in creative design methods [3], it is possible to imagine potential PGMs that meet the needs of a specific design challenge. However, in this manner, it is difficult to imagine all of the conceivable PGM kinematic structures for any design challenge. Only after all possible kinematic structures of PGMs have been analyzed can the appropriate PGM for a specific design challenge be chosen [4].

The conceptual design of PGMs is accomplished by listing all of the alternatives of a kinematic chain with the goal of creating an atlas that can be used as a source of suggestions for PGMs designers. The kinematic structure of a PGM involves vital information such as the number of links, whether the links are linked to other links and the category of joints that connect them. PGMs have a kinematic structure that is distinguished by the presence of only revolute and geared joints. An abstract graph representation is easily used to represent the kinematic structure. Graphs are also used to represent PGTs. The parent graph is the graph obtained if all edges in the graph are assumed to be of the same type. The spanning-tree represents a connected graph with all of the original graph's vertices but no circuits [5]. An edge-labeled spanning-tree is referred to sometimes as an acyclic graph when the edge labels are known implicitly.

Published online: October 04 2022

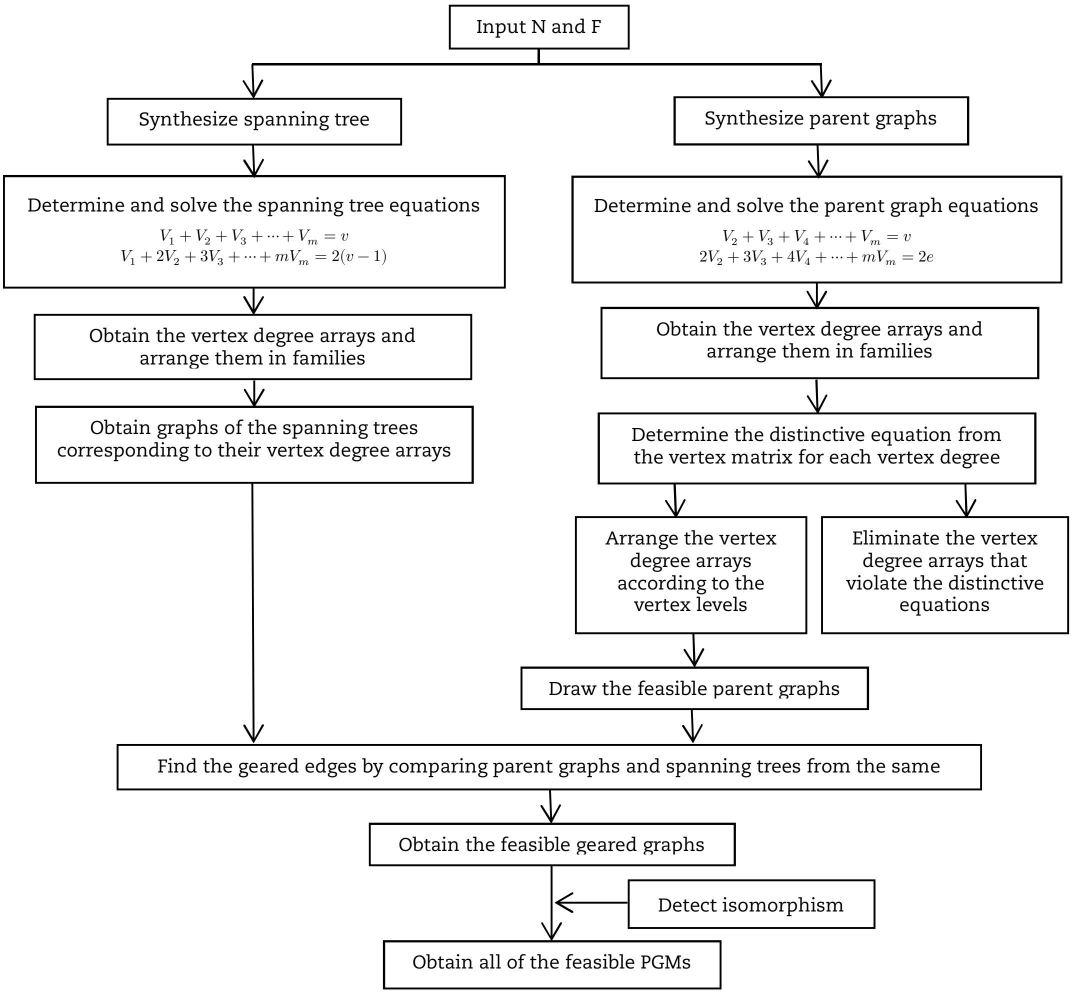

Input N and F
Synthesize spanning tree
Synthesize parent graphs
Determine and solve the spanning tree equations
$V_1+V_2+V_3+\cdots+V_m=v$
$V_1+2V_2+3V_3+\cdots+mV_m=2(v-1)$
Determine and solve the parent graph equations
$V_2+V_3+V_4+\cdots+V_m=v$
$2V_2+3V_3+4V_4+\cdots+mV_m=2e$
Obtain the vertex degree arrays and arrange them in families
Obtain the vertex degree arrays and arrange them in families
Obtain graphs of the spanning trees corresponding to their vertex degree arrays
Determine the distinctive equation from the vertex matrix for each vertex degree
Arrange the vertex degree arrays according to the vertex levels
Eliminate the vertex degree arrays that violate the distinctive equations
Draw the feasible parent graphs
Find the geared edges by comparing parent graphs and spanning trees from the same
Obtain the feasible geared graphs
Detect isomorphism
Obtain all of the feasible PGMs

Fig. 1. The primary steps of the structural synthesis method.

1.2 Formulation of the Problem

Graphs are demonstrated to be an effective tool for assisting in the synthesis of PGTs, in addition to enumerating all possible PGTs that meet strict design criteria. To derive the entire collection of geared graphs for N-links and F-DOF while utilizing the rooted graph representation [6], a method involving the use of the parent graphs and spanning trees is used. The enumeration of PGTs is broken into three steps. The initial step is to identify trees that are suitable for the construction of rooted graphs. The parent graphs are then listed. Geared graphs are detected from parent graphs and spanning trees in the third step. For a given number of vertices, the entire collection of parent graphs and spanning trees is exhaustive. As a result, it is justified that using all of the distinct parent graphs and spanning trees for generation eventually leads to the enumeration of combinatorially complete geared graphs.

To systematically enumerate kinematic structures of the exact nature, a computational technique, combinatorial analysis, and graphical approach can be used. The primary goal of this study is to present a novel method for PGTs structural synthesis. The major steps of the approach are depicted in Figure 1.

The new method avoids greatly the generation of degenerate structures and isomorphic graphs because it relies on link assortment arrays that are already different. To facilitate the automation of the methodology, various matrix representations of graphs were introduced. This approach is both analytical and algorithmic, and it appears to be promising for identifying all possible PGTs.

1.3 Literature Survey

PGT modeling employs graph-based techniques, including contour graphs, admissible graphs, acyclic graphs, and bond graphs [7-11]. Buchsbaum and Freudenstein [12] are the first to use graphs to represent the structure of PGTs, as well as to identify the notion of transfer vertex and the fundamental properties that the graph of a PGT must possess. In a later paper [13], Freudenstein found the relationship between the rotation and displacement graphs. Chatterjee and Tsai [14, 15] represented PGTs using a graph notation that precluded the formation of pseudo-isomorphic graphs. Hsu and Lam [16, 17] proposed a graph model for PGTs that differentiates between multiple and simple revolute joints. As a result, all pseudo isomorphic graphs are represented by the same graph. Yang et al. [18] suggested a graph model with solid and hollow vertices to avoid pseudo isomorphic graphs entirely.

The objective of the structural synthesis of PGTs was to catalogue all displacement graphs [19-22] for N-links and F-DOF. PGTs are typically synthesized using one of two methods: recursive [23-29] or non-recursive [12, 21, 30-32]. The non-recursive method is based on either the parent graph [12, 30, 31] or the acyclic graph of a PGT [21, 32, 33]. The problem of identifying transfer vertices is one of the challenges encountered by the traditional non-recursive method, which is typically difficult to automate. The recursive method derived n(n-1) unlabeled graphs with (n+1) vertices from the generic graph, a known unlabeled graph with (n) vertices [23]. However, since some graphs cannot be derived using the recursive method [21, 34, 35], the complete collection of graphs cannot be generated. By assigning geared pairings to a parent graph, the parent graph approach is utilized to build geared graphs. The work of [18] is more or less restricted to the inspection process and computer algorithms and does not depend on mathematical theories. The addition of geared pairs to acyclic graphs enables the acyclic graph-based technique to create displacement graphs of PGTs directly. In [36], the various algorithms utilized in the structural synthesis of PGTs are discussed. The enumeration results obtained using the aforementioned methods are inconclusive [34]. As a result, a more straightforward method for finding the entire set of combinatorially complete displacement graphs is required.

In all previous works, the synthesis process of PGTs involved methods to detect and eliminate isomorphic graphs [12, 13, 18-20, 27, 37]. Two graphs are isomorphic if their vertices and edges have a 1-1 correspondence and their incidences are preserved. Since vertices and edges are involved in the definition of isomorphism, they are usually used in the graph isomorphism test. Therefore, two graphs are isomorphic, if their vertices and edges preserves adjacency properties.

1.4 Scope and Contributions

PGTs structural synthesis can contribute to the development of new transmission systems. The majority of synthesis techniques are restricted to 1-DOF PGTs with less than ten links. However, the results of the synthesis are incompatible [34, 36]. The synthesis results are far from complete because the discrepancy in the results has yet to be resolved. By merging spanning trees with parent graphs, a new technique for methodically synthesizing PGTs is established. Using link assortment arrays, the spanning trees of an N-link, F-DoF PGT are first enumerated. Secondly, all the v-vertex parent graphs are counted. By comparing the parent graphs to the spanning trees and coloring the mismatched edges differently, the geared edges are identified. The method is built on algebraically representing the graphs through their vertex matrices. The matrix representation makes it easier to implement graphs on a digital computer. Thus, the entire set of 1-DOF geared graphs with five links is successfully constructed. A novel algorithm called trail and graph marking is proposed for first renumbering the graphs to be tested, and then identifying the isomorphism by comparing the corresponding geared strings.

The approach of Shanmukhasundaram et al. [20] that uses the conventional PGT graphs yields pseudo-isomorphic graphs; thus, according to Pathapati et al. [35], the number of graphs in [31] exceeds the number of graphs believed to be exact. Because the new model is based on the rooted graph representation, we can entirely avoid pseudo-isomorphic graphs.

Previous publications show that when the number of links is 5 or less, the synthesis results of 1-DOF PGTs are consistent. Thirteen displacement graphs are enumerated in references [12, 18, 23, and 32], to name a few. But in fact, the 5-link PGT is relatively simple and not convincing. PGTs with more than 5 links can be studied, but the object here is to present the method in as simple a form as possible. A comparison with the current results validates the accuracy of the 5-link graph results. In Refs. [18, 19, and 34], for example, the PGTs are represented by two types of graphs: graphs with at least one hollow vertex (multiple joint) and graphs with no hollow vertices. As a result, the results of the current method, which used rooted graphs, differed from those graphs with no hollow vertex. However, the results are identical to those graphs with one or more hollow vertices. The source of the disparity in existing synthesis results is dealt with effectively.

The following is a summary of the main novelty in the current work. A method based on two link assortment equations is introduced to precisely detect spanning trees. Transfer vertices and edge levels are inherent in spanning trees and can be detected without any computations. Another novelty of the synthesis methodology is the identification of the geared edges by setting the difference between parent graphs and spanning trees of the same family. We call the process which maps the spanning tree into a graph a genetic compatibility. Under the genetic compatibility hypothesis, a connected graph may have several spanning trees. The exact results of the 5-link geared graphs have been confirmed as 24.

1.5 Structure of the Paper

This paper has the following structure: Section 2 briefly reviews the fundamentals of graph representation, including the rooted graph, parent graph and spanning tree. Later the vertex matrix of the rooted graph is defined as well as how it can be partitioned by vertex levels, and how the matrix equation could be used to synthesize all potential graphs as detailed in Section 3. Section 4 details with the synthesis method for spanning trees corresponding to all link assortments. Section 5 details with the synthesis method for parent graphs. Section 6 is dedicated to genetic compatibility; geared graphs of PGTs can be synthesized by considering the parent graphs and spanning trees separately. Given a geared graph vertex degree listing, the graphs for each vertex degree listing must be synthesized first, followed by all possible combinations of the parent graphs and spanning trees, which become the graphs synthesized for the given graph vertex degree listing. Section 7 proposes a complete isomorphism identification method, including numbering rules, identification logic, and the identification process. Section 8 provides a numerical example of the essential steps of the new methodology. Finally, in section 9, the work done in the paper is summarized, certain conclusions are formed, and some potential future research directions are outlined.

2. Kinematic Structure of PGTs

In a graph representation of PGTs, edges correspond to joints and vertices to links. For example, Figure 2(c) illustrates a graphical representation of the PGM shown in Figure 2(a). The double-lined edges in Figure 2(c) represent gear pairs, while the single-lined edges represent turning pairs. The rooted graph is similarly described, with one exception: the fixed link representing the root is denoted by symbol 0. Figure 2(a) depicts the functional representation of the Ravigneaux PGT with the fixed link added to the geared kinematic chain depicted in Figure 2(b). Figure 2(d) depicts a rooted graph representation of the Ravigneaux PGM depicted in Figure 2(a). The single-line edges are labeled as a, b, and c based on their spatial location.

Figure 2(d) illustrates how the vertices of a rooted graph are divided into levels. Depending on the number of single-line edges between the vertex and the root, it is referred to as the first, second, or third-level vertex, and so on. A single-line edge connects a vertex at one level to a vertex at another level. Single-line edges of the same label all stem from the same lower-level vertex [3]. PGMs with links distributed higher than the second level are impractical in terms of mechanical complexity because they contain one or more floating carriers. Also, to simplify the conceptualization, only the vertices distributed up to the third level are considered.

Figure 2(e) depicts the parent graph of the PGM depicted in Figure 2(a). Figure 2(f) shows how to create a rooted spanning tree from a geared graph by removing all of the geared edges. The problem of identifying transfer vertices is one of the difficulties encountered in the traditional non-recursive method, which is generally difficult to automate. In contrast, the transfer vertices in the newly developed method can be easily identified using a spanning tree. A transfer vertex is any vertex of a spanning tree (other than the root vertex) that is incident with at least two single-line edges. Vertex-4 is the transfer vertex in the 6-vertex spanning tree in Figure 2(f), and thus it is the carrier. When a geared edge is inserted into a tree, only one circuit is formed. The collection of all circuits relative to a spanning tree is used to build a collection of fundamental circuits. The rooted graph, shown in Figure 2(d), contains eight vertices, three vertex levels, seven revolute edges, and five geared edges. Five fundamental circuits exist: 0-1-7-4-0, 0-2-7-4-0, 4-7-6, 0-3-6-4-0, and 0-5-6-4-0. The two revolute edges linked to the ground vertex with the same label constitute a single joint.

Following is a definition for the vertex matrix of the rooted graph:

$$Q = [q_{i,j}]_{v\times v} = \begin{cases} 1 & if\ vertex\ i\ is\ connected\ to\ vertex\ j\ by\ an\ edge \\ d_i & if\ i\ =\ j\ and\ d\ is\ the\ degree\ of\ vertex\ i \\ 0 & otherwise \end{cases} \tag{1}$$

In graph theory, the vertex matrix Q can be expressed as follows:

$$Q = A + D = BB^T \tag{2}$$

where A represents the adjacency matrix, D represents the vertex degree matrix, B represents the incidence matrix and B^T represents the transpose of the incidence matrix B. The vertex matrix can be partitioned into m^2 submatrices Q_{11}, Q_{12}, …$Q_{1\mathrm{m}}$, … Q_{21}, … Q_{mm} according to the m^{th}-vertex level as following:

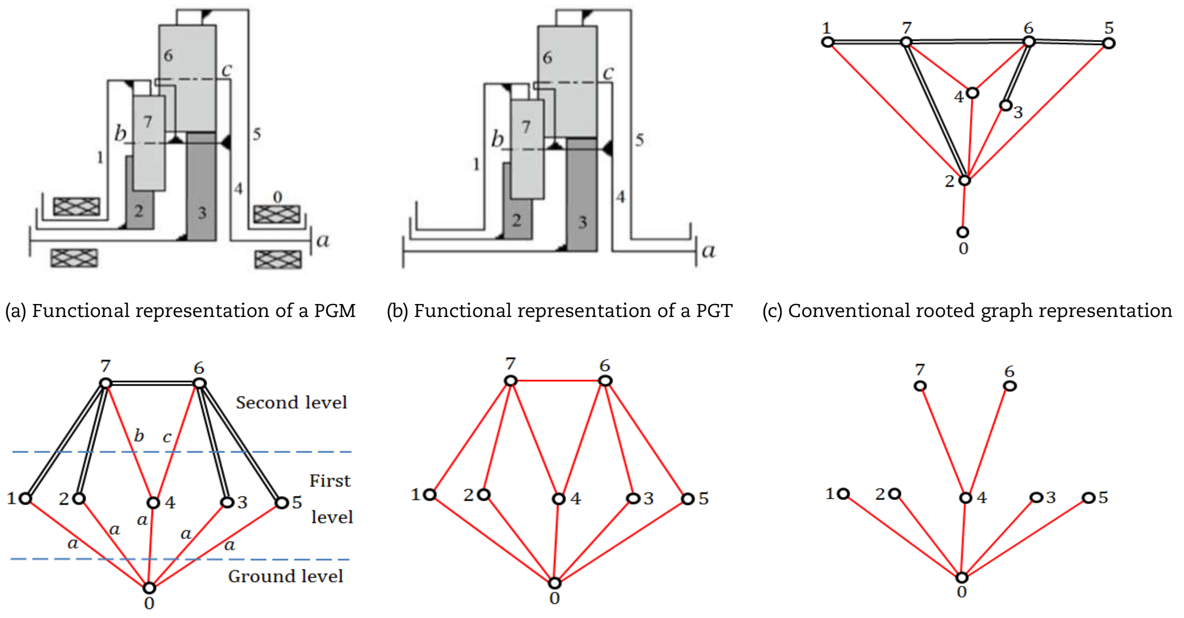
$$Q = \begin{bmatrix} Q_{11} & \vdots & Q_{12} & \vdots & Q_{13} & \vdots & \cdots & Q_{1m} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \cdots \\ Q_{21} & \vdots & Q_{22} & \vdots & Q_{23} & \vdots & \cdots & Q_{2m} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \cdots \\ Q_{31} & \vdots & Q_{32} & \vdots & Q_{33} & \vdots & \cdots & Q_{3m} \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_{m1} & \vdots & Q_{m2} & \vdots & Q_{m3} & \vdots & \cdots & Q_{mm} \end{bmatrix} \tag{3}$$

For instance, the portioned vertex matrix of the graph depicted in Figure 2(e) is as follows:

$$Q = \begin{bmatrix} & v_0 & \vdots & v_1 & v_2 & v_3 & v_4 & v_5 & \vdots & v_6 & v_7 \\ v_0 & 5 & \vdots & 1 & 1 & 1 & 1 & 1 & \vdots & 0 & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots \\ v_1 & 1 & \vdots & 2 & 0 & 0 & 0 & 0 & \vdots & 0 & 1 \\ v_2 & 1 & \vdots & 0 & 2 & 0 & 0 & 0 & \vdots & 0 & 1 \\ v_3 & 1 & \vdots & 0 & 0 & 2 & 0 & 0 & \vdots & 1 & 0 \\ v_4 & 1 & \vdots & 0 & 0 & 0 & 3 & 0 & \vdots & 1 & 1 \\ v_5 & 1 & \vdots & 0 & 0 & 0 & 0 & 2 & \vdots & 1 & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots \\ v_6 & 0 & \vdots & 0 & 0 & 1 & 1 & 1 & \vdots & 4 & 1 \\ v_7 & 0 & \vdots & 1 & 1 & 1 & 0 & 0 & \vdots & 1 & 4 \end{bmatrix} \tag{4}$$

From Eq. (4), we can deduce the structural characteristics associated with the vertex matrix Q:

1. The vertex matrix Q is a symmetric matrix in which the sum of all off-diagonal elements in the i^{th} row equals the i^{th} diagonal element.
2. The i^{th} diagonal element represent the vertex degree.
3. The off-diagonal elements in the Q matrix represent the number of edges connecting two vertices.
4. The Q_{11} submatrix is a one element matrix corresponding to the degree of the root vertex $(Q_{11} = d_{root})$.



(a) Functional representation of a PGM (b) Functional representation of a PGT (c) Conventional rooted graph representation

(d) Rooted graph representation (e) Parent rooted graph (f) Rooted spanning tree

Fig. 2. A seven-link Ravigneaux planetary gear train.

5. The number of second level vertices v_{sl} (and higher) equals the total number of vertices minus the first level vertices v_{fl} along with the root vertex which can be written as:

$$v_{sl} = v - v_{fl} - v_{root} \tag{5}$$

6. Since all first-level vertices are linked to the ground-level vertex but not to one another, their sum equals the vertex degree of the root.

$$v_{fl} = d_{root} \tag{6}$$

Also, all elements of the sub-matrix Q_{22} are zero except for the diagonal elements. Since the second-level vertices are not adjacent to the ground-level vertex, all elements of Q_{13} and Q_{31} are zero.

Let $q_{i,j}$ represent the (i, j) element of Q, and keep in mind that $q_{i,j}$ corresponds to 1 or 0, $q_{i,i}$ corresponds to the vertex degree d_i and $q_{i,j}$ equals $q_{j,i}$, then the potential graphs can be generated using the matrix equation below.

$$Q = \begin{bmatrix} & v_0 & \vdots & v_1 & v_2 & \cdots & v_{fl} & \vdots & \cdots & v_{N-1} & v_N \\ v_0 & d_0 & \vdots & 1 & 1 & \cdots & 1 & \vdots & \cdots & 0 & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ v_1 & 1 & \vdots & d_1 & 0 & \cdots & 0 & \vdots & \cdots & q_{1,N-1} & q_{1,N} \\ v_2 & 1 & \vdots & 0 & d_2 & \cdots & 0 & \vdots & \cdots & q_{2,N-1} & q_{2,N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 & \vdots & \cdots & \cdots & \cdots \\ v_{fl} & 1 & \vdots & 0 & 0 & \vdots & d_{fl} & \vdots & \cdots & q_{fl,N-1} & q_{fl,N} \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \cdots & \cdots \\ v_{N-1} & 0 & \vdots & q_{1,N-1} & q_{2,N-1} & \vdots & q_{fl,N-1} & \vdots & \vdots & d_{N-1} & q_{N-1,N} \\ v_N & 0 & \vdots & q_{1,N} & q_{2,N} & \vdots & q_{fl,N} & \vdots & \vdots & q_{N-1,N} & d_N \end{bmatrix} \tag{7}$$

$$\begin{cases} d_1 = 1 + \cdots + q_{1,N-1} + q_{1,N} \\ d_2 = 1 + \cdots + q_{2,N-1} + q_{2,N} \\ \qquad \cdots \\ d_{fl} = 1 + \cdots + q_{fl,N-1} + q_{fl,N} \end{cases} \tag{8}$$

$$\cdots$$

$$\begin{cases} d_{N-1} = q_{1,N-1} + q_{2,N-1} + \cdots + q_{fl,N-1} + \cdots + q_{N-1,N} \\ d_N = q_{1,N} + q_{2,N} + \cdots + q_{fl,N} + \cdots + q_{N-1,N} \end{cases} \tag{9}$$

$$\begin{cases} d_1 + d_2 + \cdots + d_{fl} = d_0 + d_N & For\ N = fl + 1 \\ d_1 + d_2 + \cdots + d_{fl} = d_0 + d_N + d_{N-1} + \cdots + 2 \times q_{N-1,N} + \cdots & For\ N > fl + 1 \end{cases} \tag{10}$$

3. Systematic Synthesis Methodology

The degrees of freedom F of a PGT are determined by equation $F = 3(N-1) - 2J_r - 1 \times J_g$, where N is the number of links and $J_r = N - 1$ and $J_g = N - 1 -$ F are the number of revolute and geared joints, respectively. A v-vertex, F-DOF rooted graph contains (N + 1) vertices, (v – 2 – F) geared edges and (v – 1) revolute edges. Therefore, the sum of all the edges is:

$$e = 2v - F - 3 \tag{11}$$

and

$$v = N + 1 \tag{12}$$

and the number of fundamental circuit or loops is:

$$L = 1 + e - v \tag{13}$$

One can classify links in a PGT based on the number of edges they contain. Whether a vertex has two, three, or four edges determines whether it is binary, ternary, or quaternary. If V_i is the number of vertices with i edges, then $[V_2, V_3, \dots, V_m]$ is a link assortment array of a kinematic chain. As a result, the following equations can be used to derive all possible link assortments for a rooted parent graph with v-vertices.

$$V_2 + V_3 + V_4 + \cdots + V_m = v \tag{14}$$

$$2V_2 + 3V_3 + 4V_4 + \cdots + mV_m = 2e \tag{15}$$

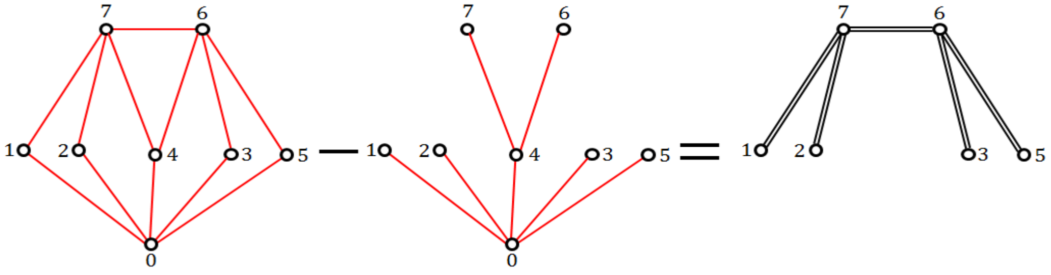
where $m = L + 1$ is the maximal vertex degree. Corresponding to all of the link assortments, all the parent graphs of F-DOF PGTs with N-links are first synthesized.

A rooted spanning tree is a subgraph of a rooted parent graph that contains all the vertices of a graph, despite the edges being distinct. Every rooted graph has a minimum of one spanning tree. The tree depicted in Figure 2(f) has multiple pendant vertices (a degree one vertex is a pendant vertex). In a v-vertex tree, there are (v-1) edges, and thus 2(v-1) degrees must be distributed among the v vertices.

All possible link assortments for a rooted spanning tree with v-vertex can be derived from the following equations:

$$V_1 + V_2 + V_3 + \cdots + V_m = v \tag{16}$$

$$V_1 + 2V_2 + 3V_3 + \cdots + mV_m = 2(v-1) \tag{17}$$



(a) Parent graph (b) Spanning tree (c) geared edges

Fig. 3. Detection of the geared edges from the difference between the parent graph and the spanning tree.

Equations (16) and (17) can be solved for all possible combinations of the link assortment arrays $[V_1, V_2, V_3, V_4, \ldots, Nm]$ of a rooted spanning tree where V_i represent the number of vertices with i edges. For the rooted spanning tree shown in Figure 2(f), a particular solution to these equations is found to be $V_1 = 6$, $V_3 = 1$, and $V_5 = 1$. This solution is known as link assortment and is represented by [6, 0, 1, 0, 1], and its vertex degree array is [5, 3, 1, 1, 1, 1, 1]. Corresponding to a spanning tree, the edges of a rooted parent graph can be subdivided into two subgraphs, one containing revolute edges and the other containing geared edges. The revolute edges include all edges that comprise the spanning tree, whereas the geared edges include all edges that do not comprise the spanning tree. The revolute edges are those that form the spanning tree, whereas the geared edges are those that do not. Hence, geared graphs can be generated from parent graphs and spanning trees by setting the difference between them as geared edges. Also, the union of the spanning trees and geared edges constitutes the rooted geared graphs.

Relying on their parent graphs and spanning trees, a novel algorithm for structural synthesis of PGTs is proposed:

Step 1. Enumerate all the v-vertex spanning trees.

Step 2. Enumerate all the vertex degree arrays of the parent graphs.

Step 3. Arrange the vertices according to their degrees and levels, and eliminate the vertex degree arrays that violate the distinctive equations.

Step 4. Enumerate all the v-vertex parent graphs.

Step 5. Determine the geared edges by setting the difference between parent graphs and spanning trees, as shown in Fig. 3.

Step 6. Repeat Step 5 until all the v-vertex spanning trees have been used, then the graphs of N-link F-DOF PGTs will be created.

Table 1. Graphs of the spanning trees corresponding to the vertex degree arrays.

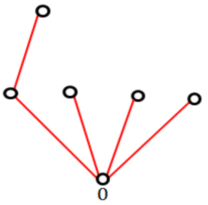
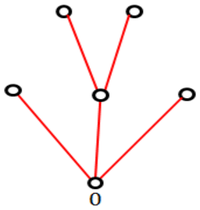
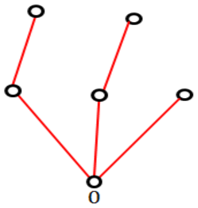
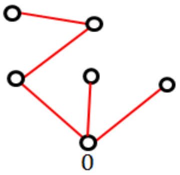
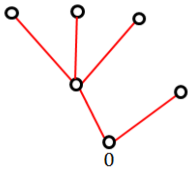
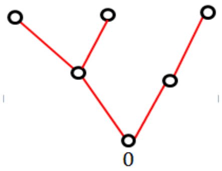
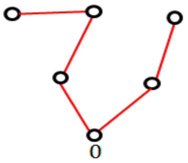
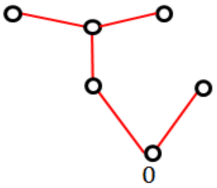
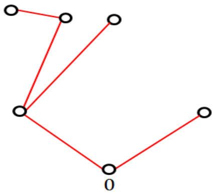
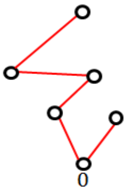
Family 1	Family 2	
[4, 2, 1, 1, 1, 1]	[3, 3, 1, 1, 1, 1]	[3, 2, 2, 1, 1, 1]
		
		[3, 2, 2, 1, 1, 1]
		
Family 3		
[2, 4, 1, 1, 1, 1]	[2, 3, 2, 1, 1, 1]	[2, 2, 2, 2, 1, 1]
		
[2, 2, 3, 1, 1, 1]	[2, 3, 2, 1, 1, 1]	[2, 2, 2, 2, 1, 1]
		

Table 2. Graphs of the spanning trees suitable for double-planet PGTs.

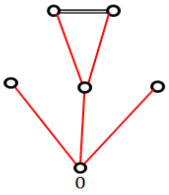
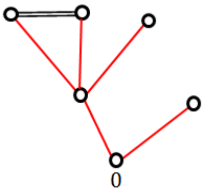
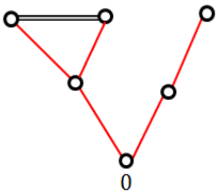
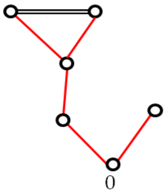
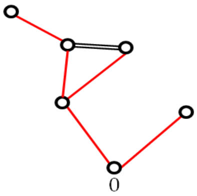
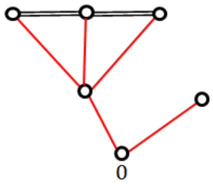
Family 2	Family 3			
[3, 3, 1, 1, 1, 1]	[2, 4, 1, 1, 1, 1]	[2, 3, 2, 1, 1, 1]	[2, 2, 3, 1, 1, 1]	[2, 3, 2, 1, 1, 1]
				

Table 3. Graphs of the spanning trees suitable for triple-planet PGTs.

Family 3
[2, 4, 1, 1, 1, 1]


4. Enumeration of Spanning Trees

The new synthesis method begins by listing all of the v-vertex spanning trees that can be used to create rooted graphs. Equations (16) and (17) can be solved for all possible combinations of $[V_1, V_2, V_3, V_4, \ldots, Nm]$.

Since PGTs with revolute and gear joints are the concern of this work, the number of vertices v and edges e can be calculated from Eqs. (11) and (12), respectively. For a PGT with 5 links ($N = 5$) and 1-DOF ($F = 1$), its number of vertices v is 6 and its number of edges e is 8. The number of revolute edges e_r is 5, the number of geared edges e_g is 3, and the maximal degree of a vertex m is 4. Equations (16) and (17) can be written as:

$$V_1 + V_2 + V_3 + V_4 = 6 \tag{18}$$

$$V_1 + 2V_2 + 3V_3 + 4V_4 = 10 \tag{19}$$

For a 6-vertex spanning tree, four link assortments can be acquired from Eqs. (18) and (19), namely; [4, 1, 0, 1], [4, 0, 2, 0], [3, 2, 1, 0], and [2, 4, 0, 0]. The first digit in the link array represents the number of vertices with degree one, the second the number of vertices with degree two, and so on. The vertex degree array of a spanning tree is a list of integers representing the vertex degrees in descending order. The vertex degree arrays corresponding to the above link assortments are: [4, 2, 1, 1, 1, 1], [3, 3, 1, 1, 1, 1], [3, 2, 2, 1, 1, 1], and [2, 2, 2, 2, 1, 1]. Table 1 shows the graphs of the ten spanning trees corresponding to these vertex degree arrays. The vertex degree array corresponding to each spanning tree is listed above it.

The new method can be used to create a spanning tree atlas with v-vertices. In the following sections, we will not be limited to graphs with a vertex distribution beyond the second level. Hence, the graph corresponding to level 3 and 4 will be discussed. Graphs of the spanning trees suitable for double-planet PGTs (two vertices in the same level connected by a geared edge) are shown in Table 2. Some geared edges are incorporated into the spanning trees to draw the reader's attention to them.

Table 3 shows graphs of the spanning trees suitable for triple-planet PGTs (three vertices in the same level connected by geared edges).

5. Parent Graph Enumeration

Equations (14) and (15) can be solved for all possible combinations of the v-vertex parent graphs $[V_2, V_3, V_4, \ldots, Nm]$. For $v = 6$ and $e = 8$, we get:

$$V_2 + V_3 + V_4 = 6 \tag{20}$$

$$2V_2 + 3V_3 + 4V_4 = 16 \tag{21}$$

Three link assortment arrays can be acquired from Eqs. (20) and (21), namely; [4, 0, 2], [3, 2, 1], and [2, 4, 0]. The vertex degree arrays corresponding to the above link assortments are: [4, 4, 2, 2, 2, 2], [4, 3, 3, 2, 2, 2], and [3, 3, 3, 3, 2, 2].

For the purpose of classification, vertex degree arrays are grouped into families. Each family is distinguished by vertex degree arrays with the same vertex degree at the root, and not necessarily the maximum.

Family 1: [4, 4, 2, 2, 2, 2], [4, 3, 3, 2, 2, 2].
Family 2: [3, 4, 3, 2, 2, 2], [3, 3, 3, 3, 2, 2].
Family 3: [2, 4, 4, 2, 2, 2], [2, 4, 3, 3, 2, 2], [2, 3, 3, 3, 3, 2].

5.1 Identification of Parent Graphs By Distinctive Equations

The third step is to arrange the vertex degree arrays according to the vertex levels and eliminate the arrays that violate the distinctive equations that can be obtained from the vertex degree arrays enumerated in the second step. Since $v = 6$, $v_{root} = 1$ and $v_{fl} = d_{root}$, we can calculate the number of the second level vertices and above for each family from Eq. ($v_{sl} = v - v_{fl} - v_{root}$) in the following way:

Family 1: $v_{fl} = \ d_{root} = 4$, hence, $\ v_{sl} = 5 - 4 = 1$

Family 2: $v_{fl} = \ d_{root} = 3$, hence, $\ v_{sl} = 5 - 3 = 2 = 2 + 0 = 1 + 1$

Case 1 (up to the second level): $v_{sl} = 2$

Case 2 (up to the third level): $v_{sl} = 1 + 1$

Family 3: $v_{fl} = \ d_{root} = 2$, hence, $\ v_{sl} = 5 - 2 = 3 = 3 + 0 = 2 + 1 = 1 + 2 = 1 + 1 + 1$

From Eq. (10), the vertex matrix for family 1 can be written as follows:

$$Q = \begin{bmatrix} & v_0 & \vdots & v_1 & v_2 & v_3 & v_4 & \vdots & v_5 \\ v_0 & d_0 & \vdots & 1 & 1 & 1 & 1 & \vdots & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots \\ v_1 & 1 & \vdots & d_1 & 0 & 0 & 0 & \vdots & q_{1,5} \\ v_2 & 1 & \vdots & 0 & d_2 & 0 & 0 & \vdots & q_{2,5} \\ v_3 & 1 & \vdots & 0 & 0 & d_3 & 0 & \vdots & q_{3,5} \\ v_4 & 1 & \vdots & 0 & 0 & 0 & d_4 & \vdots & q_{4,5} \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots \\ v_5 & 0 & \vdots & q_{1,5} & q_{2,5} & q_{3,5} & q_{4,5} & \vdots & d_5 \end{bmatrix} \tag{22}$$

From Eqs. (10) the distinctive equation for family 1 is:

$$d_1 + d_2 + d_3 + d_4 = d_0 + d_5 \tag{23}$$

subject to the constraints that $d_1, d_2, d_3, d_4 \ \leq 2$, $d_5 \leq 4$ and $d_0 = 4$.

Any vertex degree array that satisfies the distinctive equation of family one represents a feasible solution. For the vertex degree array [4, 4, 2, 2, 2, 2], we get: $d_0 = 4, d_1 = 2, d_2 = 2, d_3 = 2, d_4 = 2$, and $d_5 = 4$. The second vertex degree array ([4, 3, 3, 2, 2, 2]) should be excluded because it does not satisfy the distinctive equation. Hence, the first vertex degree array is the only feasible solution. Therefore, there is only one parent graph, as shown in column 1 of Table 4. The vertex matrix for family 2 (up to the second level) can be written as follows:

$$Q = \begin{bmatrix} & v_0 & \vdots & v_1 & v_2 & v_3 & \vdots & v_4 & v_5 \\ v_0 & d_0 & \vdots & 1 & 1 & 1 & \vdots & 0 & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots \\ v_1 & 1 & \vdots & d_1 & 0 & 0 & \vdots & q_{1,4} & q_{1,5} \\ v_2 & 1 & \vdots & 0 & d_2 & 0 & \vdots & q_{2,4} & q_{2,5} \\ v_3 & 1 & \vdots & 0 & 0 & d_3 & \vdots & q_{3,4} & q_{3,5} \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots \\ v_4 & 0 & \vdots & q_{1,4} & q_{2,4} & q_{3,4} & \vdots & d_4 & q_{4,5} \\ v_5 & 0 & \vdots & q_{1,5} & q_{2,5} & q_{3,5} & \vdots & q_{4,5} & d_5 \end{bmatrix} \tag{24}$$

From Eqs. (10) the distinctive equation for family 2 is:

$$d_1 + d_2 + d_3 = d_0 + d_4 + d_5 + 2q_{4,5} \tag{25}$$

subject to the constraints that $d_1, d_2, d_3 \ \leq 3$, $\ d_4, d_5 \leq 4$ and $d_0 = 3$. When $q_{4,5} = 0$, no edge connects the two vertices of the second-level and the distinctive equation becomes:

$$d_1 + d_2 + d_3 = d_0 + d_4 + d_5 \tag{26}$$

For the vertex degree array [3, 3, 3, 3, 2, 2], we get: $d_0 = 3, d_1 = 3, d_2 = 3, d_3 = 2, d_4 = 3$, and $d_5 = 2$. As a result, we obtain another parent graph as shown in column 2 of Table 4.

When $q_{4,5} = 1$, an edge connects the two vertices of the second level and the distinctive equation becomes:

$$d_1 + d_2 + d_3 = d_0 + d_4 + d_5 - 2 \tag{27}$$

For the vertex degree array [3, 3, 3, 3, 2, 2], we get: $d_0 = 3, d_1 = 3, d_2 = 2, d_3 = 2, d_4 = 3$, and $d_5 = 3$. As a result, we obtain another parent graph as shown in column 3 of Table 4.

For the vertex degree array [3, 4, 3, 2, 2, 2], we get: $d_0 = 3, d_1 = 3, d_2 = 2, d_3 = 2, d_4 = 2$, and $d_5 = 4$. As a result, we obtain another parent graph as shown in column 4 of Table 4. The vertex matrix for family 2 (up to the third level) can be written as follows:

$$Q = \begin{bmatrix} & v_0 & \vdots & v_1 & v_2 & v_3 & \vdots & v_4 & \vdots & v_5 \\ v_0 & d_0 & \vdots & 1 & 1 & 1 & \vdots & 0 & \vdots & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ v_1 & 1 & \vdots & d_1 & 0 & 0 & \vdots & q_{1,4} & \vdots & q_{1,5} \\ v_2 & 1 & \vdots & 0 & d_2 & 0 & \vdots & 0 & \vdots & q_{2,5} \\ v_3 & 1 & \vdots & 0 & 0 & d_3 & \vdots & 0 & \vdots & q_{3,5} \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ v_4 & 0 & \vdots & q_{1,4} & 0 & 0 & \vdots & d_4 & \vdots & 1 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ v_5 & 0 & \vdots & q_{1,5} & q_{2,5} & q_{3,5} & \vdots & 1 & \vdots & d_5 \end{bmatrix} \tag{28}$$

The distinctive equation can be obtained easily from the vertex degrees as:

$$d_1 + d_2 + d_3 = 3 + q_{1,4} + q_{1,5} + q_{2,4} + q_{3,4} \tag{29}$$

$$d_4 + d_5 = 2 + q_{1,4} + q_{1,5} + q_{2,4} + q_{3,4} \tag{30}$$

From Eqs. (29) and (30), we get:

$$d_4 + d_5 = d_1 + d_2 + d_3 - 1 \tag{31}$$

Subject to the constraints that $d_2, d_3, d_5 \; \leq 2$, $d_1 \leq 3$, $d_4 \leq 4$ and $d_0 = 3$.

For the vertex degree array [3, 4, 3, 2, 2, 2], we get: $d_0 = 3, d_1 = 3, d_2 = 2, d_3 = 2, d_4 = 2$, and $d_5 = 4$. As a result, we obtain the same parent graph as that shown in column 4 of Table 4. The vertex matrix for family 3 can be written as follows:

$$Q = \left[\begin{array}{ccccccccc} & v_0 & \vdots & v_1 & v_2 & \vdots & v_3 & v_4 & v_5 \\ v_0 & d_0 & \vdots & 1 & 1 & \vdots & 0 & 0 & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ v_1 & 1 & \vdots & d_1 & 0 & \vdots & q_{1,3} & q_{1,4} & q_{1,5} \\ v_2 & 1 & \vdots & 0 & d_2 & \vdots & q_{2,3} & q_{2,4} & q_{2,5} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \vdots & \cdots & \cdots & \cdots \\ v_3 & 0 & \vdots & q_{1,3} & q_{2,3} & \vdots & d_3 & q_{3,4} & 0 \\ v_4 & 0 & \vdots & q_{1,4} & q_{2,4} & \vdots & q_{3,4} & d_4 & q_{4,5} \\ v_5 & 0 & \vdots & q_{1,5} & q_{2,5} & \vdots & 0 & q_{4,5} & d_5 \end{array}\right] \tag{32}$$

From Eqs. (10) the distinctive equation for family 3 is:

$$d_1 + d_2 = d_0 + d_3 + d_4 + d_5 + 2q_{3,4} + 2q_{4,5} \tag{33}$$

subject to the constraints that $d_1, d_2,\ d_4 \leq 4$, $d_3, d_5 \leq 3$ and $d_0 = 2$. Table 4 shows the graphs of the ten feasible parent graphs satisfying the distinctive equations. The vertex degree array corresponding to each parent graphs is listed above it.

6. Geared Graphs Enumeration

The fourth step is to find the geared edges by comparing parent graphs and spanning trees from the same family. We call the process which maps the spanning tree into a graph a genetic compatibility. Under the genetic compatibility hypothesis, a connected graph may have several spanning trees.

Table 4. The feasible parent graphs $[d_0, d_1, d_2, d_3, d_4, d_5]$.

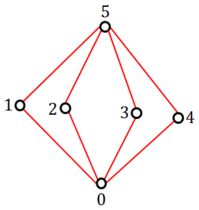
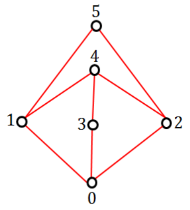
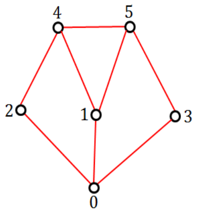
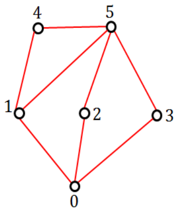
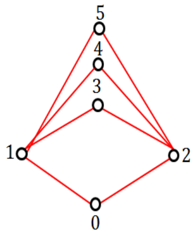
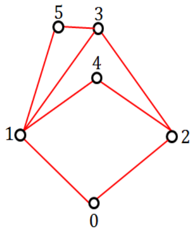
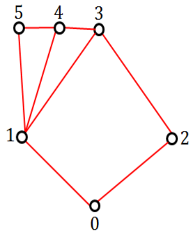
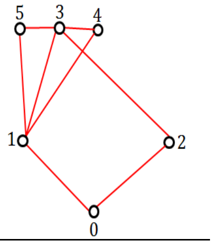
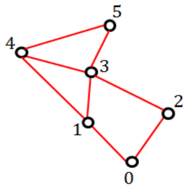
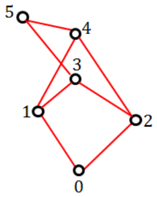
Family 1	Family 2		
[4, 2, 2, 2, 2, 4]	[3, 3, 3, 2, 3, 2]	[3, 3, 2, 2, 3, 3]	[3, 3, 2, 2, 2, 4]
			
Family 3			
[2, 4, 4, 2, 2, 2]	[2, 4, 3, 3, 2, 2]	[2, 4, 2, 3, 3, 2]	[2, 4, 2, 4, 2, 2]
			
[2, 3, 2, 4, 3, 2]	[2, 3, 3, 3, 3, 2]		
			

Table 5. The feasible geared graphs.

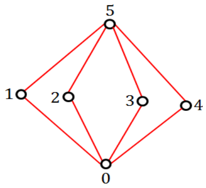
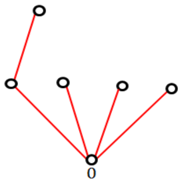
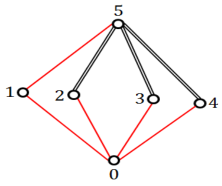
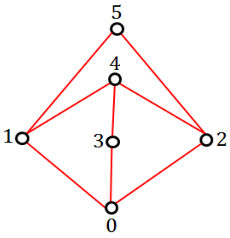
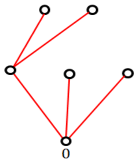
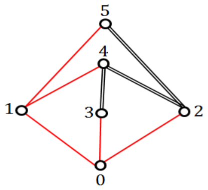
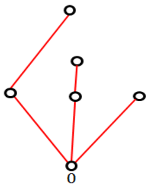
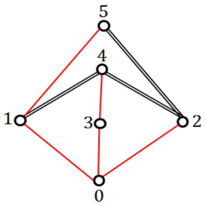
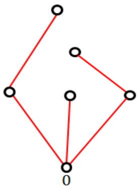
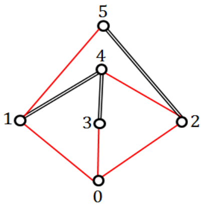
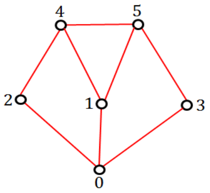
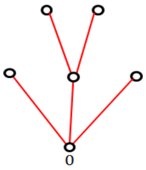
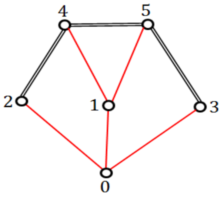
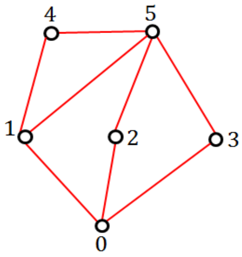
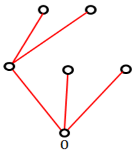
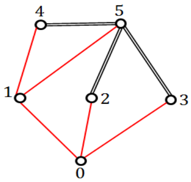
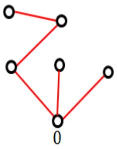
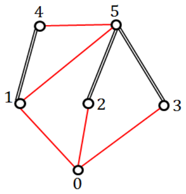
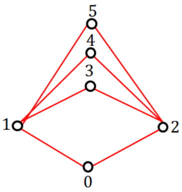
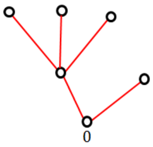
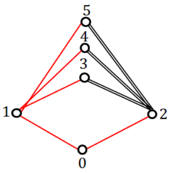
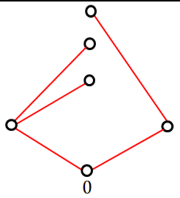
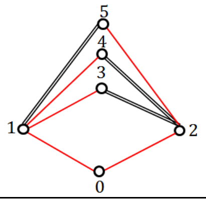
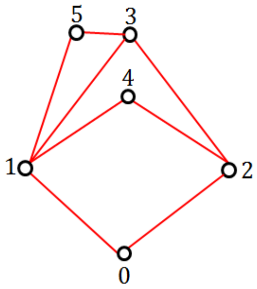
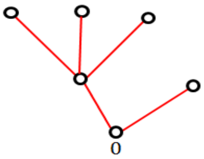
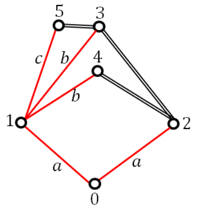
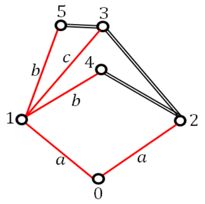
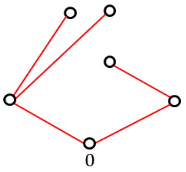
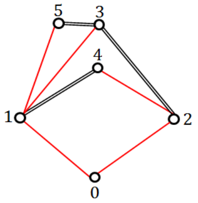
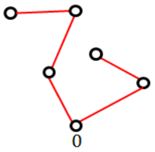
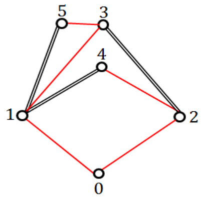
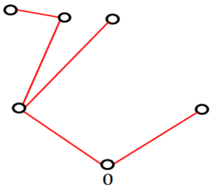
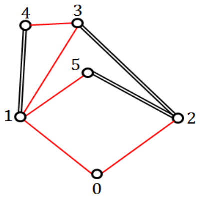
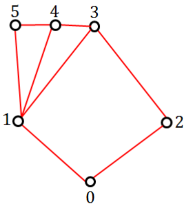
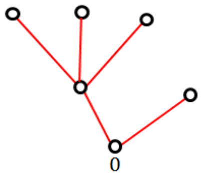
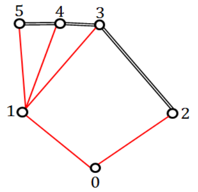
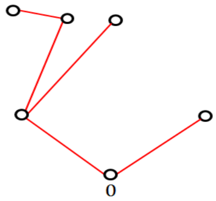
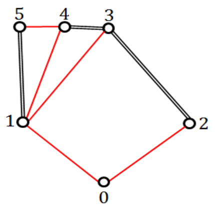
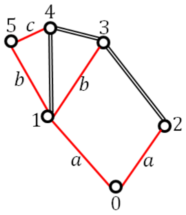
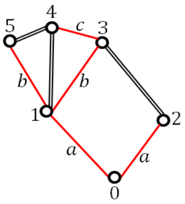
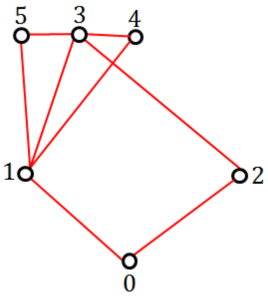
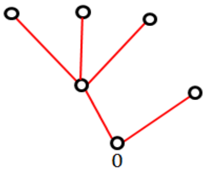
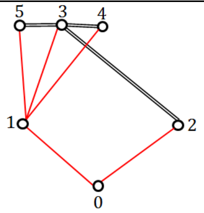
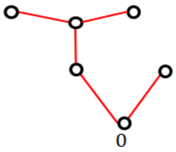
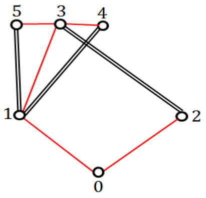
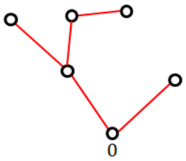
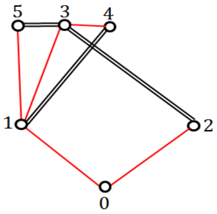
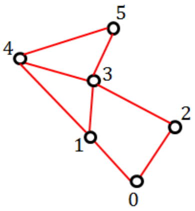
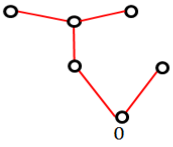
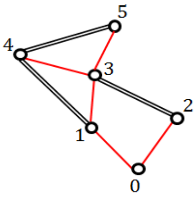
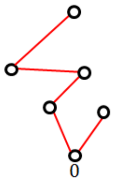
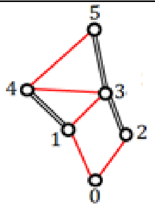
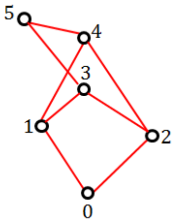
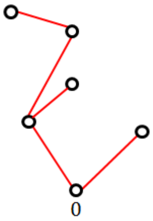
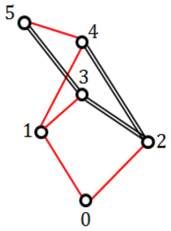
Family 1		
[4, 2, 2, 2, 2, 4]	[4, 2, 1, 1, 1, 1]	[4, 2, 2, 2, 2, 4] [4, 2, 1, 1, 1, 1]
		

Table 5. Continued.

	Family 2	
[3, 3, 3, 2, 3, 2]	[3, 3, 1, 1, 1, 1]	[3, 3, 3, 2, 3, 2] [3, 3, 1, 1, 1, 1]
		
	[3, 2, 1, 2, 1, 1]	[3- 3, 3, 2- 3, 2] [3- 2, 1, 2- 1, 1]
		
	[3, 2, 2, 1, 1, 1]	[3- 3, 3, 2- 3, 2] [3- 2, 2, 1- 1, 1]
		
[3, 3, 2, 2, 3, 3]	[3, 3, 1, 1, 1, 1]	[3, 3, 2, 2, 3, 3] [3, 3, 1, 1, 1, 1]
		
[3, 3, 2, 2, 2, 4]	[3, 3, 1, 1, 1, 1]	[3, 3, 2, 2, 2, 4] [3, 3, 1, 1, 1, 1]
		
	[3, 2, 1, 1, 1, 2]	[3, 3, 2, 2, 2, 4] [3, 2, 1, 1, 1, 2]
		
	Family 3	
[2, 4, 4, 2, 2, 2]	[2, 4, 1, 1, 1, 1]	[2, 4, 4, 2, 2, 2] [2, 4, 1, 1, 1, 1]
		
	[2, 3, 2, 1, 1, 1]	[2, 4, 4, 2, 2, 2] [2, 3, 2, 1, 1, 1]

		
[2, 4, 3, 3, 2, 2]	[2, 4, 1, 1, 1, 1]	[2, 4, 3, 3, 2, 2] [2, 4, 1, 1, 1, 1]
		
		[2, 4, 3, 2, 2, 3] [2, 4, 1, 1, 1, 1]
		
	[2, 3, 2, 1, 1, 1]	[2, 4, 3, 3, 2, 2] [2, 3, 2, 1, 1, 1]
		
	[2, 2, 2, 2, 1, 1]	[2, 4, 3, 3, 2, 2] [2, 2, 2, 2, 1, 1]
		
	[2, 3, 1, 2, 1, 1]	[2, 4, 3, 3, 2, 2] [2, 3, 1, 2, 1, 1]
		
[2, 4, 2, 3, 3, 2]	[2, 4, 1, 1, 1, 1]	[2, 4, 2, 3, 3, 2] [2, 4, 1, 1, 1, 1]
		
	[2, 3, 1, 1, 2, 1]	[2, 4, 2, 3, 3, 2] [2, 3, 1, 1, 2, 1]

		
		[2, 3, 1, 1, 2, 1] [2, 4, 2, 3, 2, 3]
		
		[2, 3, 1, 1, 2, 1] [2, 4, 2, 3, 2, 3]
		
[2, 4, 2, 4, 2, 2]	[2, 4, 1, 1, 1, 1]	[2, 4, 2, 4, 2, 2] [2, 4, 1, 1, 1, 1]
		
	[2, 2, 1, 3, 1, 1]	[2, 4, 2, 4, 2, 2] [2, 2, 1, 3, 1, 1]
		
	[2, 3, 1, 2, 1, 1]	[2, 4, 2, 4, 2, 2] [2, 3, 1, 2, 1, 1]
		
[2, 3, 2, 4, 3, 2]	[2, 2, 1, 3, 1, 1]	[2, 3, 2, 4, 3, 2] [2, 2, 1, 3, 1, 1]
		
	[2, 2, 1, 2, 2, 1]	[2, 3, 2, 3, 4, 2] [2, 2, 1, 2, 2, 1]

		
[2, 3, 3, 3, 3, 2]	[2, 3, 1, 1, 2, 1]	[2, 3, 1, 1, 2, 1] [2, 3, 3, 3, 3, 2]
		

The precise results of the 5-link geared graphs are confirmed to be 24. The synthesis results in this study do not match those in the literature. One possible explanation is that some PGTs cannot be generated from graphs with no multiple joint. All of the methods that rely on the parent graph and the acyclic graph generate graphs without multiple joints. According to the findings of this paper, every graph has at least one multiple joint that is designated as a root vertex. Therefore, the results of this paper are greater than the previous ones.

7. Identification of Isomorphism

Graphs are classified into families depending on their link assortment arrays. Of course, graphs from various families cannot be isomorphic. The vertex degree arrays of parent graphs and spanning trees provide the necessary condition to check graph isomorphism. If the degrees of vertices of two graphs are not equivalent, they are not isomorphic because there is no one-to-one correspondence between their vertices. If they are equivalent, there exists a finite number of isomorphic possibilities. The finite number of isomorphic possibilities and geared strings provide the sufficient condition to check graph isomorphism. As a result, the trail and graph marking procedure described here provides both the required and sufficient conditions for graph isomorphism tests.

The trail and graph marking procedure to check graph isomorphism is described below:

Step 1: Check the vertex degree array of both parent graphs, if they are not equivalent, then the two graphs are not isomorphic, if they are equivalent, proceed to step 2.

Step 2: Examine the vertex degree array of their spanning trees, if they are not equivalent, then the two graphs are not isomorphic, if they are equivalent, proceed to step 3.

Step 3: Determine the weighted vertex degree of each vertex in the graph. The weighted vertex degree is obtained by assigning edge weights of one for revolute edges and two for geared edges.

Step 4: Find a trail, which is a series of edges connecting all of the graph's vertices. Vertex and edge repetition is permitted.

Step 5: Find the weighted vertex degree array of the trail found in step 4.

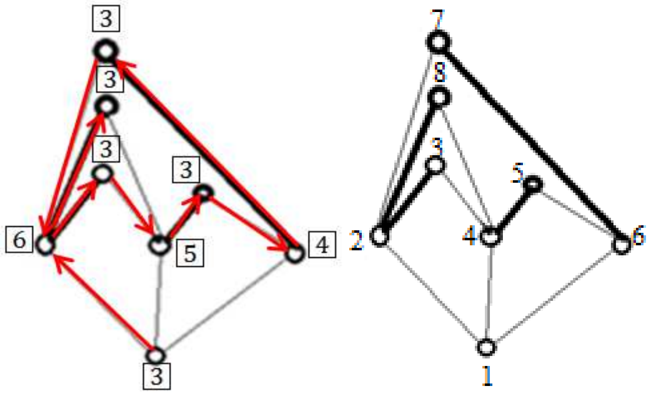
Step 6: Number the vertices of the trail in ascending order, beginning at the root.

Step 7: Since the weighted vertex degree array of two graphs is the same, follow the trail and label the vertices of the second graph according to weighted vertex degree array found in step 5 in ascending order, beginning at the root.

Step 8: Check the numbering of the geared strings of both geared graphs, if they are not equivalent, then the two graphs are not isomorphic, if they are equivalent, then the two graphs are isomorphic.

This is accomplished by generating a list of vertex degrees for each geared graph. The first and second rows of the vertex degree listing represent the degrees of vertices of the parent graphs and spanning trees respectively.

In general, two graphs having the same vertex degree listing are not necessary to be isomorphic. This is because the vertex degree listing of a graph only shows the listing of degrees of vertices of the graph, the listing itself does not take into account the connections between vertices.



(a) (b)

Fig. 4. Numbering the graph according to the weighted vertex degree array of the trail.

Table 6. Graphs sharing the same vertex degree listing $\begin{bmatrix}3&4&4&3&2&2&2&2\\3&2&3&2&1&1&1&1\end{bmatrix}$

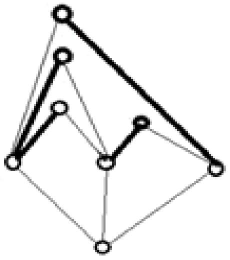
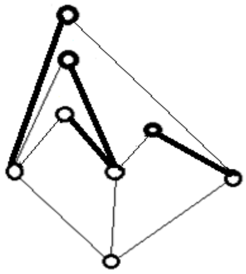
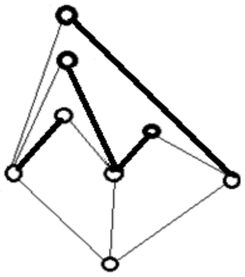
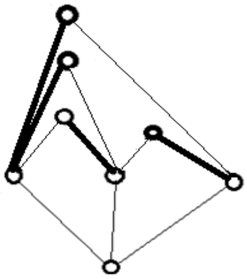
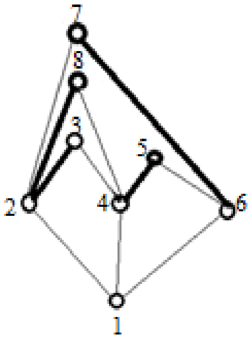
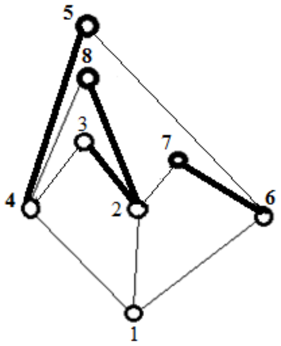
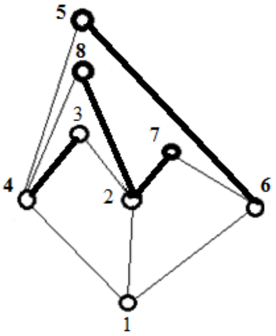
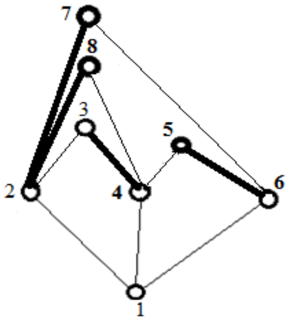
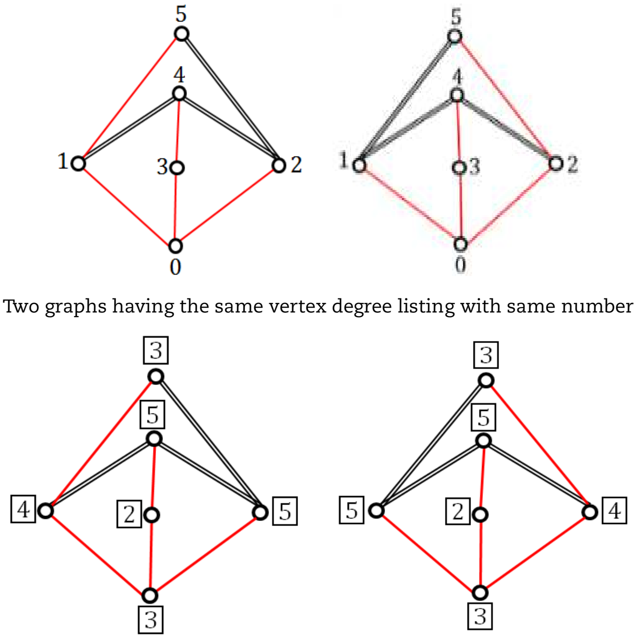
Weighted Vertex Degree= $[3\ \ 6\ \ 5\ \ 4\ \ 3\ \ 3\ \ 3\ \ 3]$ Trail $[3\ \ 6\ \ 3\ \ 5\ \ 3\ \ 4\ \ 3\ \ 6\ \ 3]$			
			
Numbering the graphs according to the weighted vertex degree array of the trail.			
			
Geared Strings			
$GS_1 = [3,2,8]\ [4,5]\ [6,7]$	$GS_2 = [3,2,8]\ [4,5]\ [6,7]$	$GS_3 = [3,4][7,2,8]\ [5,6]$	$GS_4 = [3,4][7,2,8]\ [5,6]$
(a)	(b)	(c)	(d)

Table 6 shows four graphs sharing the same vertex degree listing $\begin{bmatrix}3&4&4&3&2&2&2&2\\3&2&3&2&1&1&1&1\end{bmatrix}$, but there are isomorphic possibilities, as explained below.

For the four graphs in Table 6, the weighted vertex degree array is $[3\ \ 6\ \ 5\ \ 4\ \ 3\ \ 3\ \ 3\ \ 3]$. For example, in Figure 4(b) edges e_{12}, e_{23}, e_{27}, and e_{28} are incident with vertex 2. The weights of the four edges are 1, 2, 1, and 2, respectively. As a result, vertex 2 has a weighted vertex degree of $2 + 1 + 1 + 2 = 6$.

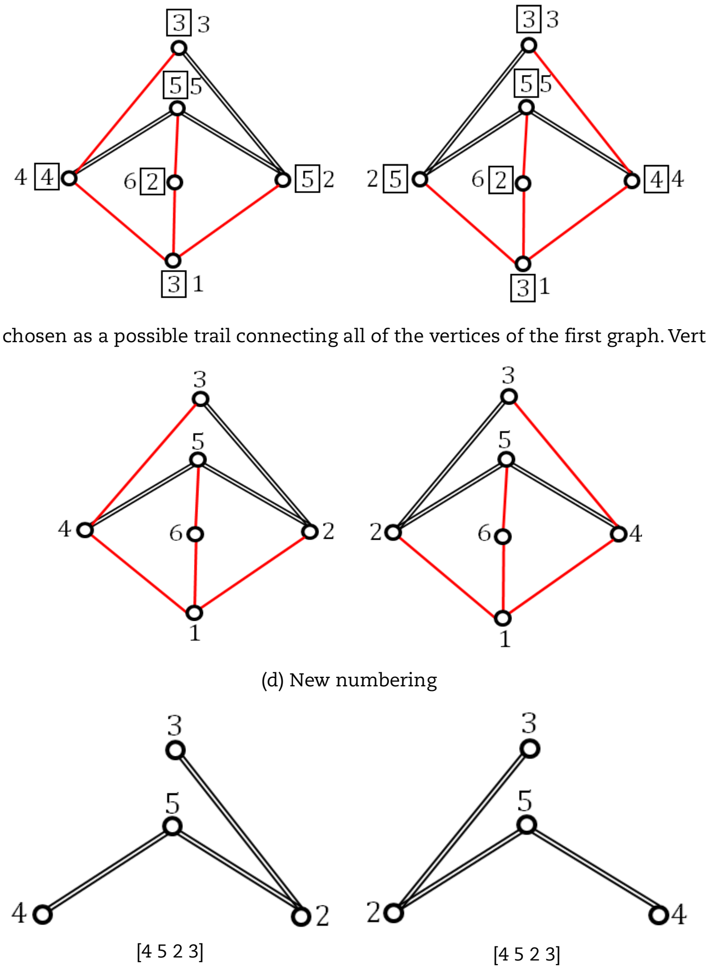
The trail $[3\ \ 6\ \ 3\ \ 5\ \ 3\ \ 4\ \ 3\ \ 6\ \ 3]$ is chosen as a possible trail connecting all of the vertices of the first graph in Table 6 and is depicted in Figure 4(a). Any other trail can be chosen. The numbered graph is shown in Figure 4(b)



(a) Two graphs having the same vertex degree listing with same numbering

(b) Graphs with vertices labeled with the weighted vertex degree.

Fig. 5. The steps of the trail and graph marking procedure for identifying isomorphism for two graphs with the same vertex degree listing $\begin{bmatrix}3&3&3&3&2&2\\3&2&2&1&1&1\end{bmatrix}$.



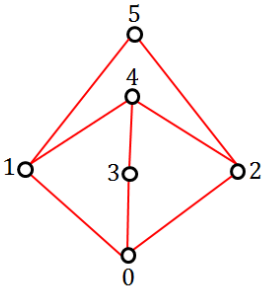
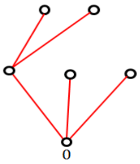
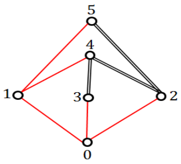
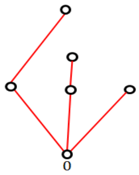
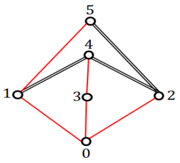
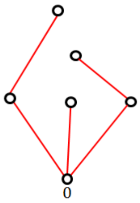
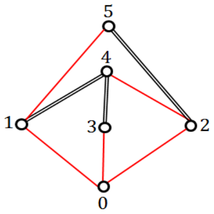
(c) The trail [3 5 3 4 5 2] is chosen as a possible trail connecting all of the vertices of the first graph. Vertices are numbered accordingly.

(d) New numbering

(e) Geared Strings

Fig. 5. Continued.

Table 7. The feasible geared graphs for a numbered parent graph.

Family 2		
[3, 3, 3, 2, 3, 2]	[3, 3, 1, 1, 1, 1]	[3, 3, 3, 2, 3, 2] [3, 3, 1, 1, 1, 1]
		
	[3, 2, 1, 2, 1, 1]	[3, 3, 3, 2, 3, 2] [3, 2, 1, 2, 1, 1]
		
	[3, 2, 2, 1, 1, 1]	[3, 3, 3, 2, 3, 2] [3, 2, 2, 1, 1, 1]
		

The geared strings of the four graphs are designated as GS_1, GS_2, GS_3, and GS_4, respectively. We have $GS_1 = [3,2,8]\ [4,5]\ [6,7]$, $GS_3 = [3,4][7,2,8]\ [5,6]$, $GS_1 = GS_2$, and $GS_3 = GS_4$. The two graphs in columns (a) and (b) are isomorphic because $GS_1 = GS_2$. In addition, the graphs in columns (c) and (d) are isomorphic. Even though the four graphs in Table 6 have the same vertex degree listing, the two graphs in columns (a) and (c) are not isomorphic due to different geared strings. Furthermore, the graphs in columns (b) and (d) are not isomorphic. References [38, 39] corroborate this result.

Figure 5 depicts the steps of the trail and graph marking procedure for identifying isomorphism for two graphs with the same vertex degree listing $\begin{bmatrix} 3 & 3 & 3 & 3 & 2 & 2 \\ 3 & 2 & 2 & 1 & 1 & 1 \end{bmatrix}$. The two graphs are isomorphic because the geared strings are satisfied.

Graphically and mathematically, structural isomorphism can be identified by the adjacency properties of geared graphs. The adjacency properties of a parent graph are determined by the spanning tree contained within the graph or by the arrangement of the geared edges. For example, the geared graphs in Table 7 represent different gear trains but use the same labeling. The difference is due to the spanning trees in the graphs. This avoids the time-consuming and complex process required by other methods. The main advantages of this method are compact notations, high accuracy, and ease of use. Another advantage is that the method eliminates the vast majority of isomorphic structures automatically.

8. Demonstrative Example

For a PGT with five links ($N = 5$) and 1-DOF ($F = 1$), from Eq. (12), the number of vertices v is 6, from Eq. (11), the number of edges e is 8, from Eq. (13), the number of fundamental circuit or loops L is 3, and the maximal degree of a vertex $m = L + 1 = 4$.

For a 6-vertex spanning tree, four link assortment arrays can be acquired from Eqs. (18) and (19), namely; [4, 1, 0, 1], [4, 0, 2, 0], [3, 2, 1, 0], and [2, 4, 0, 0]. For example, a spanning tree with $\mathrm{V}_1 = 4$, $\mathrm{V}_2 = 1$, $\mathrm{V}_3 = 0$ and $\mathrm{V}_4 = 1$ has a link assortment array of [4, 1, 0, 1], and a vertex degree array of [4, 2, 1, 1, 1, 1] (or there are four vertices with degree one, one with degree two, and one with degree four). The vertex degree arrays corresponding to the above four link assortments are: [4, 2, 1, 1, 1, 1], [3, 3, 1, 1, 1, 1], [3, 2, 2, 1, 1, 1], and [2, 2, 2, 2, 1, 1].

For the purpose of classification, vertex degree arrays are grouped into families. Each family is designated by d_{root}.

Family 1 ($d_{root} = 4$): [4, 2, 1, 1, 1, 1].

Family 2 ($d_{root} = 3$): [3, 3, 1, 1, 1, 1], [3, 2, 2, 1, 1, 1].

Family 3 ($d_{root} = 2$): [2, 4, 1, 1, 1, 1], [2, 3, 1, 1, 1, 1], [2, 2, 2, 2, 1, 1].

Given a vertex degree array, it is possible to enumerate all the spanning trees.

Since all first-level vertices are connected to the root but not to one another, their total number equals the vertex degree of the root. For the vertex degree array [4, 2, 1, 1, 1, 1] where $d_0 = d_{root} = 4, d_1 = 2, d_2 = 1, d_3 = 1, d_4 = 1,$ and $d_5 = 1$. The number of first level vertices v_{fl} equals the vertex degree of the root d_0. From Eq. (6), $v_{fl} = d_{root} = 4$, and from Eq. (5), the number of second level vertices v_{sl} equals the total number of vertices minus the first level vertices v_{fl} along with the root vertex which can be written as: $v_{sl} = v - v_{fl} - v_{root} = 6 - 4 - 1 = 1$.

Figure 6 illustrates the spanning tree corresponding to the vertex degree array [4, 2, 1, 1, 1, 1]. There are 4 vertices in the first level (v_1, v_2, v_3, v_4), 1 vertex in the second level (v_5), and the root vertex of degree four (v_0) in the ground level.

Transfer vertices are readily apparent from a spanning tree. A transfer vertex is any vertex of a spanning tree (other than the root vertex) that is incident with at least two single-line edges. For the spanning tree depicted in Figure 6, vertex 1 is the transfer vertex because it is the only vertex (other than the root vertex) that is incident with two single-line edges.

Using Eqs. (14) and (15), all link assortment arrays for parent graphs can be determined. For $v = 6$ and $e = 8$, three link assortment arrays can be acquired from Eqs. (20) and (21), namely; [4, 0, 2], [3, 2, 1], and [2, 4, 0]. The vertex degree arrays corresponding to these link assortments are: [4, 4, 2, 2, 2, 2], [4, 3, 3, 2, 2, 2], and [3, 3, 3, 3, 2, 2].

The vertex degree arrays are grouped into families:

Family 1 ($d_{root} = 4$) : [4, 4, 2, 2, 2, 2], [4, 3, 3, 2, 2, 2].

Family 2 ($d_{root} = 3$) : [3, 4, 3, 2, 2, 2], [3, 3, 3, 3, 2, 2].

Family 3 ($d_{root} = 2$) : [2, 4, 4, 2, 2, 2], [2, 4, 3, 3, 2, 2], [2, 3, 3, 3, 3, 2].

The most important step in the new method is to arrange the vertex degree arrays according to the vertex levels and eliminate the arrays that violate the distinctive equations. Parent graphs and spanning trees are classified into families depending on their link assortment arrays. Of course, parent graphs and spanning trees from various families cannot be compatible. The vertex degree arrays of parent graphs and spanning trees with the same root vertex degree provide the necessary condition for checking graph compatibility. Only family 1 will be utilized to illustrate the new method.

The vertex matrix for family 1 can be constructed based on the structural characteristics of vertex matrix Q.

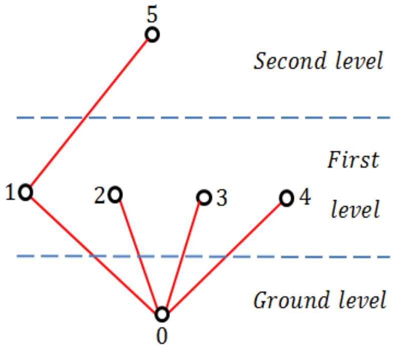


Fig. 6. Spanning tree corresponding to the vertex degree array [4, 2, 1, 1, 1, 1].

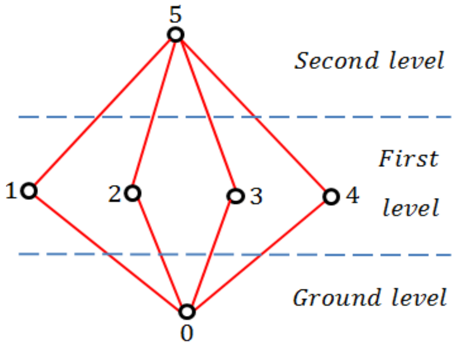
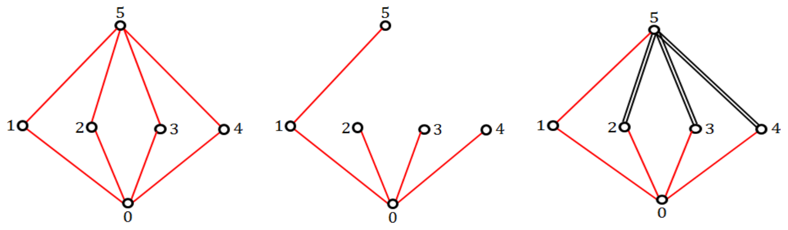
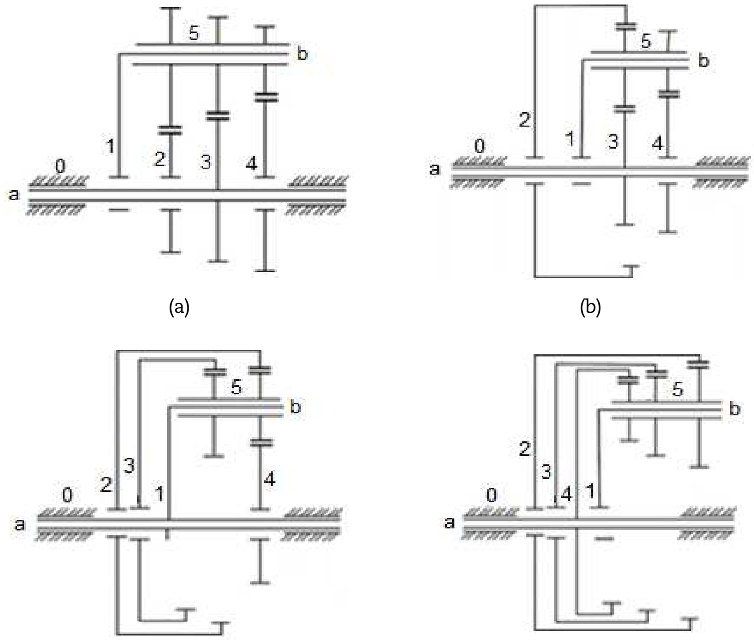


Fig. 7. The feasible parent graph corresponding to the vertex degree array [4; 2, 2, 2, 2; 4].



(a) Parent graph (b) Spanning tree (c) geared graph

Fig. 8. Detection of the geared edges from the difference between the parent graph and the spanning tree.



(a) (b)

(c) (d)

Fig. 9. Functional diagrams that correspond to Figure 7(c).

From Eq. (7), we can write Eq. (22). From Eqs. (8) and (9) the distinctive equation for family 1 can be obtained as follows:

$$\begin{cases} d_1 = 1 + 1 = 2 \\ d_2 = 1 + q_{2,5} \\ d_3 = 1 + q_{3,5} \\ d_4 = 1 + q_{4,5} \end{cases} \tag{34}$$

$$d_5 = 1 + q_{2,5} + + q_{3,5} + q_{4,5} \tag{35}$$

From Eqs. (34) and (35), we get

$$d_1 + d_2 + d_3 + d_4 = d_0 + d_5 \tag{36}$$

subject to the constraints that $d_2, d_3, d_4 \le 2, d_5 \le 4$ and $d_0 = 4$.

Any vertex degree array that satisfies the distinctive equation of family one represents a feasible solution. For the vertex degree array [4, 4, 2, 2, 2, 2], we get: $d_0 = 4, d_1 = 2, d_2 = 2, d_3 = 2, d_4 = 2$, and $d_5 = 4$. There are 4 vertices of degree two in the first level (v_1, v_2, v_3, v_4), 1 vertex of degree four in the second level (v_5), and the root vertex of degree four (v_0). First-level vertices correspond to coaxial sun (ring) gears, or carriers, whereas second-level vertices correspond to planet gears. Figure 6 depicts the parent graph associated with the vertex degree array [4; 2, 2, 2, 2; 4].

The second vertex degree array [4, 3, 3, 2, 2, 2] should be excluded because it does not satisfy the distinctive equation. Hence, the first vertex degree array is the only feasible solution. Therefore, there is only one parent graph, as shown in Figure 7. The revolute edges are those that form the spanning tree, whereas the geared edges are those that do not. Hence, the geared graph can be generated from the parent graph (shown in Figure 8(a)) and the spanning tree (shown in Figure 8(b)) by setting the difference between them as geared edges.

For the spanning tree depicted in Figure 8(b), vertex 1 is the transfer vertex and is thus the planet gear carrier. For the geared graph depicted in Figure 8(c), vertices 2, 3, and 4 are coaxial sun gears or ring gears, whereas vertex 5 is the planet gear. Figure 9 depicts functional diagrams that are compatible with Figure 8(c).

Figure 9(c) is a functional schematic diagram of the Minuteman cover drive, a well-known transmission. In this transmission, the input, output, and fixed links are the sun gear, link 4, the first ring gear, link 2, and the second ring gear, link 3. The compound planet, link 5, is supported by a revolving joint on link 1, the carrier. It meshes with sun gear number 4 and both ring gears. The joints connecting link 4 to links 1, 2, and 3 are revolute joints. On the whole, it forms a compound PGT with a 1-DOF.

9. Conclusion

Kinematic structures with the same number of degrees of freedom and links can be represented by graphs, configured with combinatorial configuration analysis, and enumerated with computer algorithms. The following are the research contributions of this work:

1. The rooted graph definition is improved to make the graph representation appropriate for graphs with and without multiple joints.
2. A novel method for synthesizing 1-DOF PGTs is described. Trees suitable for constructing rooted graphs are first identified. The parent graphs are then listed. Finally, geared graphs are discovered by inspecting their parent graphs and spanning trees. This approach is both analytical and algorithmic, and it appears to be promising for identifying all possible PGTs. The new approach avoids greatly the generation of degenerate structures and isomorphic graphs because it relies on link assortment arrays that are already different. The current genetic compatibility method is anticipated to be used in the future to synthesize PGTs with more DOF and links.
3. A novel method based on two link assortment equations is introduced to precisely detect spanning trees. Transfer vertices and edge levels are inherent in spanning trees and are detected without the use of any computations.
4. This work develops the vertex matrix of the rooted graph whose distinctive equation is used to arrange the vertex degree arrays according to the vertex levels and eliminate the arrays that violate the distinctive equations.
5. The precise results of the 5-link geared graphs are confirmed to be 24. In Refs. [18, 19, 34], for example, the PGTs are represented by two types of graphs: graphs with at least one hollow vertex (multiple joint) and graphs with no hollow vertices. As a result, the results of the current method, which used rooted graphs, differed from those graphs with no hollow vertex. However, the results are identical to those graphs with one or more hollow vertices. The source of the disparity in existing synthesis results is dealt with effectively.
6. According to the findings of this paper, every graph has at least one multiple joint that is designated as a root vertex. As a result, the results of this paper outperform the previous ones, and PGTs that cannot be generated using other methods are identified. The literature is examined for possible explanations for the contradictory synthesis results.
7. A novel algorithm called trail and graph marking is proposed for first renumbering the graphs to be tested, and then identifying the isomorphism by comparing the corresponding geared strings. When identifying isomorphism, it is not necessary to test graphs from various families of parent graphs or spanning trees, which greatly improves the efficiency of detection. The algorithm is straightforward and computationally efficient. It can be modified to function with other kinematic structures.
8. Structural isomorphism can be identified uniquely because the vertex degree listing and gear string representation have a one-to-one correspondence with the graph representation. This allows for the storage of a large number of graphs on a computer for later use. The vertex degrees listing outperforms most other methods where no classification is involved and all candidate graphs must be tested.
9. The current approach can be employed to synthesize PGTs with multiple degrees of freedom.
10. The complete PGT database allows for the generation of new transmission configurations.

Author Contributions

All authors contributed equally to this work. All authors discussed the findings, reviewed, and approved the final version of the article.

Acknowledgments

Not applicable.

Conflict of Interest

The authors declared no potential conflicts of interest concerning the research, authorship, and publication of this article.

Funding

The authors received no financial support for the research, authorship, and publication of this article.

Data Availability Statements

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Nomenclature

A	The adjacency matrix	$\mathrm{a_{ij}}$	The (i,j) elements of the adjacent matrix
B	The incidence matrix	B^T	The transpose of the incidence matrix B
D	The vertex degree matrix	d_{root}	The degree of the root vertex
e	Number of edges	J_r	The number of revolute joints
F	Number of degrees of Freedom	J_g	The number geared joints
GS	Geared string	$q_{i,j}$	The (i,j) element of Q
L	Number of independent loops	$V_2, V_3, \ldots, V_m$	Number of binary, ternary…*m*-nary vertices of parent graph
m	The maximal degree of vertex	$[V_1, V_2, V_3, \ldots, V_m]$	Link assortment array for spanning tree
N	Number of Links	v_{fl}	The number of first level vertices
o	Root or the ground vertex	v_{sl}	The number of second level vertices
Q	Vertex matrix	v_{root}	The root vertex
V	Vertices	DOF	Degree of Freedom
v	Number of vertices	PGT	Planetary gear train
		PGM	Planetary gear mechanism

References

[1] Tsai, L.W., *Mechanism Design Enumeration of Kinematic Structures According to Function*, CRC Press LLC, Boca Raton, 2001.

[2] Döring, U., Brix, T., Reeßing, M., Application of computational kinematics in the digital mechanism and gear library DMG-Lib, *Mech. Mach. Theory*, 41(8), 2006, 1003-1015.

[3] Pennestrì, E., Belfiore, N.P., On Crossley's contribution to the development of graph based algorithms for the analysis of mechanisms and gear trains, *Mech. Mach. Theory*, 89, 2015, 1003-1015.

[4] Ding, H., Yang, W., Kecskeméthy, A., *Automatic Structural Synthesis and Creative Design of Mechanisms*, Springer Nature Singapore, 2022.

[5] Robin, J.W., *Introduction to Graph Theory*, Addison Wesley Longman, 1996.

[6] Esmail, E.L., Juber, A.H., An Application of Graph Theory for the Detection of Degenerate Structures in Planetary Gear Trains, *Proceedings of the ASME 2021 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Volume 8A: 45th Mechanisms and Robotics Conference (MR), Virtual, Online, August 17–19, V08AT08A025, ASME, 2021.

[7] Shanmukhasundaram, V.R., *Number synthesis and structure based rating of multilinkepicyclic gear trains satisfying gruebler's degree of freedom equation*, Ph.D. Thesis, BITS Pilani – Hyderabad Campus, India, 2020.

[8] Marciniec, A., Sobolak, M., Połowniak, P., Graphical method for the analysis of planetary gear trains, *Alex. Eng. J.*, 6, 2022, 4067-4079

[9] Drewniak, J., Kopec, J., Zawilak, S., Kinematical Analysis of Variants of Wind Turbine Drive by Means of Graphs, Graph-Based Modelling in Engineering, *Mechanism and Machine Science Series*, 42, 2016, 81-95.

[10] Drewniak, J., Zawilak, S., Linear-Graph and Contour-Graph based models of planetary gears, *J. Theor. Appl. Mech.*, 48, 2010, 415-433.

[11] Tan, W. , Wu, J., Ni, D., Yan, H., Xiang, E., Liu, S., Dynamic Modeling and Simulation of Double- Planetary Gearbox Based on Bond Graph, *Math. Probl. Eng.*, 2021, 2021, 1-14.

[12] Buchsbaum, F., Freudenstein, F., Synthesis of kinematic structure of geared kinematic chains and other mechanisms, *J. Mech.*, 5(3), 1970, 357–392.

[13] Freudenstein, F., An application of Boolean algebra to the motion of epicyclic drives, *ASME J. Eng. Ind.*, 93(1), 1971, 176–182.

[14] Chatterjee, G., Tsai, L.W., Enumeration of Epicyclic-Type Automatic Transmission Gear Trains, *SAE International Congress and Exposition*, Paper No. 941012, Transmission and Driveline Developments, SP-1032, 1994.

[15] Chatterjee, G., Tsai, L.W., Computer-aided sketching of epicyclic-type automatic transmission of gear trains, *ASME J. Mech. Design*, 118, 1996, 405–411.

[16] Hsu, C.H., Lam, K.T., Automatic analysis of the kinematic structure of planetary gear trains, *ASME J. Mech. Design*, 115(3), 1993, 631-638.

[17] Hsu, C.H., Lam, K.T., A New Graph Representation for the Automatic Kinematic Analysis of Planetary Gear Trains, *ASME J. Mech. Design*, 114, 1992, 196–200.

[18] Yang, W.J., Ding, H.F., Zi, B., Zhang, D., New graph representation for planetary gear trains, *ASME J. Mech. Design*, 140(1), 2018, 012303.

[19] Yang, W.J., Ding, H.F., Kecskeméthy, A., Automatic Structural Synthesis of Non-Fractionated 2-DOF Planetary Gear Trains, *Mech. Mach. Theory*, 155, 2021, 104125.

[20] Shanmukhasundaram, V.R., Rao, Y.V.D., Regalla, S.P., Enumeration of displacement graphs of epicyclic gear train from a given rotation graph using concept of building of kinematic units, *Mech. Mach. Theory*, 134, 2019, 393–424.

[21] Shanmukhasundaram, V.R., Rao, Y.V.D., Regalla, S.P., Varadaraju, D., Pennestrì, E., *Structural Synthesis and Classification of Epicyclic Gear Trains: An Acyclic Graph-Based Approach*, In: Rao Y.V.D., Amarnath C., Regalla S.P., Javed A., Singh K.K. (eds) Advances in Industrial Machines and Mechanisms, Lecture Notes in Mechanical Engineering, Springer, Singapore, 2021.

[22] Hussen, H.A., Esmail E.L., Al-Mayali, M.F., Structure synthesis of planetary gear trains using graph partitioning, *AIP Conference Proceedings*, 2386, 2022, 040030.

[23] Tsai, L.W., An Application of the Linkage Characteristic Polynomial to the Topological Synthesis of Epicyclic Gear Train, *ASME J. Mech. Transm. Autom. Des.*, 109(3),1987, 329-336.

[24] Tsai, L. W., Lin, C.C., The Creation of Non-fractionated Two-Degree-of-Freedom Epicyclic Gear Trains, *ASME J. Mech. Transm. Autom. Des.*, 111(4), 1989, 524-529.

[25] Kim, J.U., Kwak, B.M., Application of Edge Permutation Group to Structural Synthesis of Epicyclic Gear Trains, *Mech. Mach. Theory*, 25(5), 1990, 563-574.

[26] Hsu, C.H., A Graph Representation for the Structural Synthesis of Geared Kinematic Chains, *J. Frankl. Inst.*, 330(1), 1993, 131-143.

[27] Hsu, C.H., Lam, K.T., Yin, Y.L., Automatic Synthesis of Displacement Graphs for Planetary Gear Trains, *Math. Comput. Modelling*, 19(11), 1994, 67-81.

[28] Rao, Y.V.D., Rao, A.C., Generation of Epicyclic Gear Trains of One Degree of Freedom, *ASME J. Mech. Des.*, 130(5), 2008, 052604.

[29] Kamesh, V.V., Rao, K.M., Rao, A.B.S., Topological Synthesis of Epicyclic Gear Trains Using Vertex Incidence Polynomial, *ASME J. Mech. Des.*, 139(6), 2017, 062304.

[30] Ravisankar, R., Mruthyunjaya, T.S., Computerized Synthesis of the Structure of Geared Kinematic Chains, *Mech. Mach. Theory*, 20(5), 1985, 367-387.

[31] Shin, J.K., Krishnamurthy, S., Standard Code Technique in the Enumeration of Epicyclic Gear Trains, *Mech. Mach. Theory*, 28(3), 1993, 347-355.

[32] Hsu, C.H., Hsu, J.J., An Efficient Methodology for the Structural Synthesis of Geared Kinematic Chains, *Mech. Mach. Theory*, 32(8), 1997, 957-973.

[33] Cui, R., Ye, Z., Sun, L., Zheng, G., Wu, C., Synthesis method for planetary gear trains without using rotation graphs, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 236(2), 2022, 972-983.

[34] Yang, W.J., Ding, H.F., The complete set of one-degree-of-freedom planetary gear trains with up to nine links, *ASME J. Mech. Des.,* 141(4), 2019, 043301.

[35] Prasad Raju Pathapati, V.V.N.R., Rao, A.C., A New Technique Based on Loops to Investigate Displacement Isomorphism in Planetary Gear Trains,

ASME J. Mech. Des., 124(4), 2002, 662-675.
[36] Shanmukhasundaram, V.R., Rao, Y.V.D., Regalla, S.P., *Review of Structural Synthesis Algorithms for Epicyclic Gear Trains*, Springer, Singapore, 2021.
[37] Xu, X., Sun, H., Liu, Y., Dong, P., Matrix-Based Operation Method for Detecting Structural Isomorphism of Planetary Gear Train Structures, *ASME J. Mech. Des.*, 142(6), 2020, 063301
[38] Yang, W.J., Ding, H.F., The perimeter loop-based method for the automatic isomorphism detection in planetary gear trains, *ASME J. Mech. Des.*, 140(12), 2018, 123302.
[39] Hsu, C.H., Displacement Isomorphism of Planetary Gear Trains, *Mech. Mach. Theory*, 29, 1994, 513–523.

ORCID iD

Hind A. Nafeh https://orcid.org/ 0000-0002-6448-4628
Essam L. Esmail https://orcid.org/0000-0002-2507-786X

How to cite this article: Nafeh H.A., Esmail E.L., Abdali S.H. Automatic Structural Synthesis of Planetary Geared Mechanisms using Graph Theory, *J. Appl. Comput. Mech.*, 9(2), 2023, 384–403. https://doi.org/10.22055/jacm.2022.41255.3721

Publisher's Note Shahid Chamran University of Ahvaz remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.