Mech. Sci., 13, 371–386, 2022
https://doi.org/10.5194/ms-13-371-2022

Mechanical Sciences
Open Access

Novel loop tree for the similarity recognition of kinematic chains

Lei Wang[1,2], Liang Sun[1,2], Rongjiang Cui[3], Yadan Xu[3], Gaohong Yu[1,2], and Chuanyu Wu[1,2]

[1]Faculty of Mechanical Engineering and Automation, Zhejiang Sci-Tech University, Hangzhou 310018, China
[2]Zhejiang Province Key Laboratory of Transplanting Equipment and Technology, Hangzhou 310018, China
[3]Hangzhou Vocational and Technical College, Hangzhou 310018, China

Correspondence: Yadan Xu (xuyadan2005@163.com)

Received: 6 March 2022 – Revised: 27 March 2022 – Accepted: 28 March 2022 – Published: 25 April 2022

Abstract. The similarity recognition of kinematic chains (KCs) is helpful for improving the efficiency of configuration synthesis, which has been paid more and more attention in recent years. The existing recognition methods are divided into the definition method and feature constant method. Among them, the definition method is difficult to adopt in practice because of its long operation time, especially when the number of similar vertices in KCs is large. In this paper, the new concepts of a loop tree (LT) and a loop tree matrix (LTM) have been proposed, which improve the efficiency of similarity recognition. This method is applied on the complete structure of the following: 8-link with 1 DOF (degree of freedom), 9-link with 2 DOF, 10-link with 1 DOF, 12-link with 1 DOF, 13-link with 2 DOF, 14-link with 3 DOF, 15-link with 4 DOF planar single-joint KCs, and contracted graphs with up to six independent loops. All results are verified by the definition method to prove the good applicability, reliability, and efficiency of the proposed method. Simultaneously, the application case of the similarity recognition in a mechanism creation is given to provide a reference for an innovative design.

1 Introduction

The similarity recognition of kinematic chains (KCs) is an important part of the mechanism of innovation design, which can avoid isomorphism in the synthesis process and reduce the generation of redundant design schemes (Yan, 1992; Tsai, 2000; Sun et al., 2022). At present, relatively little research has been done on similarity recognition, including the definition method and the characteristic constant method. Among them, similarity recognition is accurate when using the definition method of graph theory, but numerous computations will be generated, especially when the number of links exceeds 10 (Sun et al., 2020). And for the characteristic constant method, most of them are not dedicated to similarity recognition but are more often applied to isomorphism recognition. Similarity recognition and isomorphism detection are essentially the same. The difference lies in that one is a recognition within graphs and the other is a recognition between graphs. Therefore, the characteristic constant method applied to isomorphism detection can also be used for similarity recognition.

Vertices in graph theory represent KC components, and a similarity analysis of the KC components can be transformed into a similarity recognition problem of vertices in topological graphs. Harary and Palmar (1966) discussed graph similarity, where it is noted that, if vertices u and v of graph G are similar, then $G-u \cong G-v$ and vice versa. Freudenstein (1967) provided the condition for similarity from the perspective of the permutation cycle of the automorphism group. However, the calculation amount is substantially larger, and a KC with n links requires $n!$ cycle times. Lv (1989) verified the necessary conditions for similar vertices. Wang and Yan (2002) classified the similarity between components into symmetrical, parallel, transferred, and irregular states. This method only uses the vertex degree of the KC and does not reflect the specific connection between the components and misjudgments in several special cases. Sun et al. (2020) proposed similarity recognition methods based on the definition of similarity. However, the judgment is complicated when the number of similar vertices in the KCs are large.

There are many methods for isomorphism discrimination, which can also be used for similarity recognition, such as the eigenvalue, eigenvector, characteristic polynomial (Uicker and Raicu, 1975; Mruthyunjaya, 1984; Mruthyunjaya and Balasubramanian, 1987; Yan and Hall, 1982; He et al., 2005; Cubillo and Wan, 2005; Sunkari and Schmidt, 2006), Hamming number-based (Rao and Varada Raju, 1991; Rao and Rao, 1993, 2008; Dharanipragada and Chintada, 2016; Sun et al., 2017), code-based (Ambekar and Agrawal, 1987; Shin and Krishnamurty, 1992; Tang and Liu, 1993; Rai and Punjabi, 2018, 2019), and distance-based or path-based (Yadav et al., 1996; Venkata Kamesh et al., 2017) methods. For example, Uicker and Raicu (1975) applied characteristic polynomials of adjacent matrices to isomorphism identification. Ambekar and Agrawal (1987) proposed a coding-based isomorphism identification method. Rao and Varada Raju (1991) first introduced the concept of the Hamming distance in the field of information and communication to the mechanism. Sunkari and Schmidt (2006) introduced the eigenvalue and eigenvector of the adjacency matrix as the criterion into the isomorphism identification method. Moreover, Venkata Kamesh et al. (2017) detected the isomorphism of linkage and geared KCs using the concept of net distance in graph theory; counterexamples have been found for 10-link KCs in this paper. By considering the simplicity of the method, scholars, such as Shin and Krishnamurty (1992), Tang and Liu (1993), and Rai and Punjabi (2018, 2019), have conducted extensive research on code-based methods. Overall, the similarity recognition of the KC, based on the graph theory definition has no advantage in efficiency, especially when the number of similar vertices in the KCs are large. Existing characteristic constant methods cannot fully express the vertices and the graph's information, and their reliability will become poor with the large number of links. And many other methods have not been applied to similarity recognition. Therefore, in this paper, an idea with high efficiency and reliability of similarity recognition is proposed using a loop tree (LT) and a loop tree matrix (LTM) method.

The rest of this paper is organized as follows. Section 2 proposes the concept of the LT and LTM. Section 3 applies the LT and LTM to similarity recognition and gives some examples. Section 4 introduces the similarity algorithm application. Section 5 analyzes the recognition results. Finally, Sect. 6 concludes the paper.

2 Concepts of LT and LTM

2.1 LT

Set A is a vertex of graph G. All loops with A as the starting vertex are searched and given in the form of a tree (where vertex A is the root of the tree). For example, in Fig. 1c, the LT with vertex 4 is as shown in Fig. 2 (thick, solid lines in LT represent parallel edges 1–5).

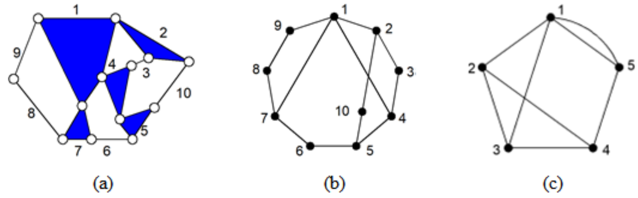


Figure 1. (a) A 10-link with 1 DOF (degree of freedom) KC, **(b)** a topological graph, and **(c)** a contracted graph.

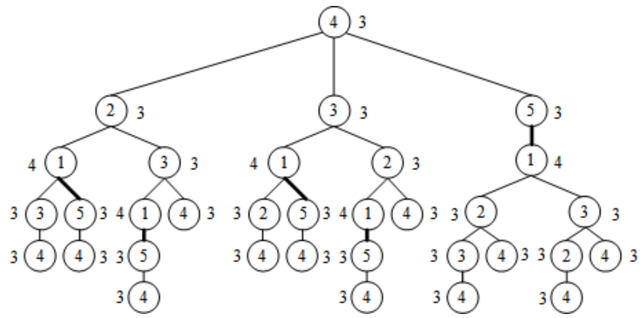


Figure 2. The LT with vertex 4 as the root of Fig. 1c.

The LT represents all the information of vertices in the graph, such as the vertex degree information, assignment information of connecting edges, and loop information. If the LTs of two vertices are the same, then the statuses of the two vertices in the graph are equivalent. For example, in Fig. 1c, the LT has vertex 1 as the root, as shown in Fig. 3. Although the tree structure is the same as that in Fig. 2, the vertex degree and the assignment information of connection edges are inconsistent; thus, vertices 4 and 1 are non-similar.

2.2 LTM

For the convenience of computer implementation, a matrix expression of LT is proposed, as shown in Table 1. Each number in a loop contains three messages. The first message is the degree of the root, the second message specifies the number of branch edges, and the third message is the value assigned for branch edges (fine and thick solid lines are assigned 1 and 2, respectively). But when a cycle is completed, there is a last number containing only one piece of information (i.e., the degree of the root). For example, the loop 4-2-1-3-4 in Fig. 2 can be expressed as 331, 321, 421, 311, and 3. Other loops can then be expressed, and the sequence of the numbers is converted into a matrix form. The number of digits is insufficient to fill 0. The LTM of Fig. 2 is shown in Eq. (1).

Definition 1. The row elements of the LTM are sorted in descending order (i.e., L_d). For example, L_d of Fig. 2 is pre-

Table 1. Elements of LT.

The loop does not return to the initial vertex			Otherwise
First message	Second message	Third message	
Degree of the root	Number of branch edges	Assigned value of branch edges	Degree of the root

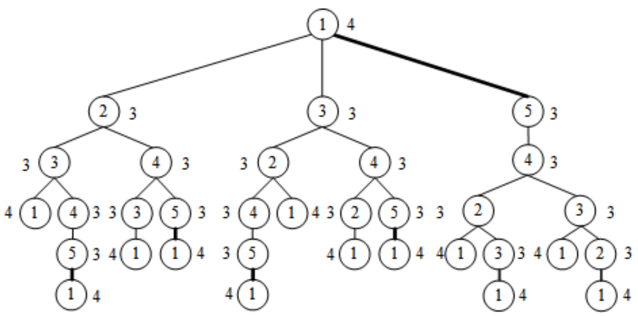


Figure 3. The LT with vertex 1 as the root of Fig. 1c.

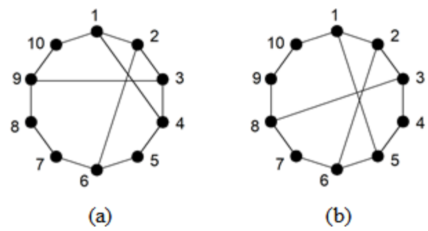


Figure 4. **(a)** A 10-link PSKC-1 and **(b)** a 10-link PSKC-2.

sented as follows:

$$\mathrm{LTM}=\begin{bmatrix} 331 & 321 & 421 & 311 & 3 & 0\\ 331 & 321 & 422 & 311 & 3 & 0\\ 331 & 321 & 321 & 412 & 311 & 3\\ 331 & 321 & 321 & 3 & 0 & 0\\ 331 & 321 & 421 & 311 & 3 & 0\\ 331 & 321 & 422 & 311 & 3 & 0\\ 331 & 321 & 321 & 412 & 311 & 3\\ 331 & 321 & 321 & 3 & 0 & 0\\ 331 & 312 & 421 & 321 & 311 & 3\\ 331 & 312 & 421 & 321 & 3 & 0\\ 331 & 312 & 421 & 321 & 311 & 3\\ 331 & 312 & 421 & 321 & 3 & 0 \end{bmatrix},$$

$$L_\mathrm{d}=\begin{bmatrix} 331 & 321 & 422 & 311 & 3 & 0\\ 331 & 321 & 422 & 311 & 3 & 0\\ 331 & 321 & 421 & 311 & 3 & 0\\ 331 & 321 & 421 & 311 & 3 & 0\\ 331 & 321 & 321 & 412 & 311 & 3\\ 331 & 321 & 321 & 412 & 311 & 3\\ 331 & 321 & 321 & 3 & 0 & 0\\ 331 & 321 & 321 & 3 & 0 & 0\\ 331 & 312 & 421 & 321 & 311 & 3\\ 331 & 312 & 421 & 321 & 311 & 3\\ 331 & 312 & 421 & 321 & 3 & 0\\ 331 & 312 & 421 & 321 & 3 & 0 \end{bmatrix}. \quad (1)$$

3 Similarity recognition

3.1 Similarity recognition algorithm

If L_d of two vertices are the same, then the two vertices are judged to be similar; otherwise, they are non-similar. The steps of the similarity recognition algorithm are as follows.

First calculate L_d of each vertex and compare the L_d parameters of two vertices. If L_d parameters of two vertices are different, then these vertices are non-similar, and the next vertices would be considered. Otherwise, these vertices are similar. The next vertices are then considered until all candidate vertices are identified. For example, vertices 2 and 3 in Fig. 1c have the same L_d as shown in Eq. (2); thus, vertices 2 and 3 are similar.

$$L_{\mathrm{d}2}=L_{\mathrm{d}3}=\begin{bmatrix} 331 & 422 & 311 & 321 & 311 & 3\\ 331 & 422 & 311 & 321 & 3 & 0\\ 331 & 421 & 321 & 311 & 3 & 0\\ 331 & 421 & 321 & 3 & 0 & 0\\ 331 & 321 & 422 & 311 & 311 & 3\\ 331 & 321 & 421 & 3 & 0 & 0\\ 331 & 321 & 321 & 411 & 3 & 0\\ 331 & 321 & 321 & 312 & 411 & 3\\ 331 & 321 & 321 & 3 & 0 & 0\\ 331 & 321 & 321 & 3 & 0 & 0\\ 331 & 321 & 312 & 421 & 311 & 3\\ 331 & 321 & 312 & 421 & 3 & 0 \end{bmatrix}. \quad (2)$$

3.2 Illustrative examples of similarity recognition

The proposed method is tested on planar single-joint KCs (PSKCs) and contracted graphs of KCs to prove its versatility. Illustrative examples are provided as follows.

3.2.1 Examples of PSKCs

Example 1. In total, two 10-link PSKCs are considered in Fig. 4a and b.

The adjacency matrices of two 10-link PSKCs in Fig. 4 are entered into the similarity recognition programs. The L_d of each vertex is solved and compared with other vertices of same graph. Similar information to that in Fig. 4 is shown in Table 2. To some extent, solving the similarity of the graphs

Table 2. Similar information to that in Fig. 4.

	Similar vertices
Fig. 4a	[1, 4] [2, 3] [5, 10] [6, 9] [7, 8]
Fig. 4b	No

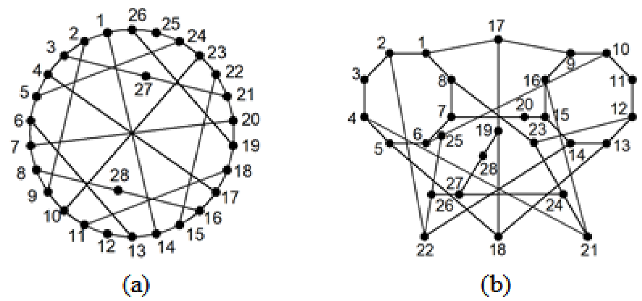


Figure 5. (a) A 28-link PSKC-1 and **(b)** a 28-link PSKC-2.

can also identify whether they are isomorphic. Because the similar information in Fig. 4a and b is different, it can be judged that they are not isomorphic.

Example 2. In total, two 28-link PSKCs are considered in Fig. 5a and b, as discussed in Uicker and Raicu (1975).

The adjacency matrices of two 28-link PSKCs in Fig. 5 are entered into the similarity recognition programs. Similar information to that in Fig. 5 is shown in Table 3. Because the similar information of Fig. 5a and b is the same, there is the possibility of isomorphism between two graphs.

3.2.2 Examples of contracted graphs

The distinction of contracted graphs lies in the existence of parallel edges, which can be distinguished by setting up the parallel edge value equal to the number of parallel edges. The similarity recognition method is same as the single-hinged KC. As shown in Fig. 8, two 10-link contracted graphs are considered.

The adjacency matrices of two 10-link contracted graphs in Fig. 6 are entered into the similarity recognition programs. Similar information to that in Fig. 6 is shown in Table 4. Because the similar information of Fig. 6a and b is different, it can be judged that they are not isomorphic.

4 Similarity algorithm application

4.1 KC similarity recognition application

In addition to the examples in Sect. 3, the similarity recognition algorithm is applied to the complete atlas of 10-link with 1 DOF PSKCs. The recognition results are consistent with the definition method of the graph theory, as shown in Appendix A (only the upper triangular elements of the adjacency matrix are given in the Appendix A).

Table 3. Similar information to that in Fig. 5.

	Similar vertices
Fig. 5a	[1, 2, 4, 5, 6, 7, 9, 10, 14, 15, 17, 18, 19, 20, 22, 23] [3, 8, 11, 13, 16, 21, 24, 26] [12, 25, 27, 28]
Fig. 5b	[1, 5, 6, 8, 9, 13, 14, 16, 17, 18, 21, 22, 23, 24, 25, 26] [2, 4, 7, 10, 12, 15, 19, 27] [3, 11, 20, 28]

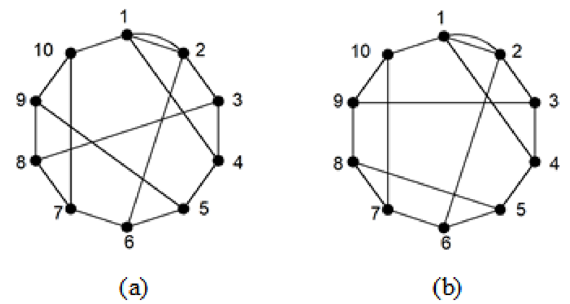


Figure 6. (a) A 10-link contracted graph (1) and **(b)** a 10-link contracted graph (2).

4.2 Innovative mechanism design

Yan (1992) proposed a regenerated KC method for innovative mechanism design, which represents logical reasoning for the KC regeneration process. In logical reasoning, the similarity of the KC components should be analyzed first to reduce redundant design schemes and improve the innovative design efficiency. Vertices in the graph theory represent KC components, and a similar analysis of the KC components can be transformed into a similarity recognition problem of vertices in topological graphs.

The working device of a loader is the main part for realizing the shoveling operation and material loading. Its structure directly affects the working size and comprehensive performance of the loader. Taking the mechanism's innovative design of a 6-link loader as an example, as shown in Fig. 7, the application of similarity recognition in mechanism innovative design is illustrated.

Step 1. Retrieve the existing design by task and enumerate all the KCs.

Figure 7c shows the original mechanism of the loader's innovative design. According to Tsai (2000), 35 9-link with 2 DOF topological graphs are obtained.

Step 2. Analyze the structural and functional constraints of the original mechanism and conducting topological graphs screening. For example, see the structural and functional constraints of Fig. 7a.

1. There must be a ternary link as a rack.

2. There must be two oil cylinders as the original moving parts, with one as the rotating oil cylinder and one as the lifting oil cylinder, where one end of the lifting oil cylinder is connected to the rack, and the other one is

Table 4. Similar information to that in Fig. 6.

	Similar vertices
Fig. 6a	No
Fig. 6b	[1, 2] [3, 4] [5, 9] [6, 10]

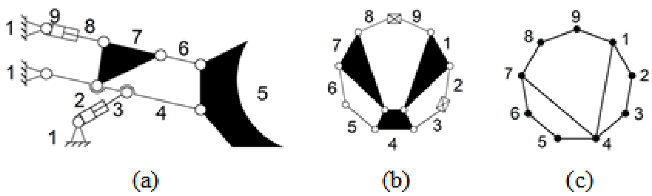


Figure 7. **(a)** Loader functional schematic, **(b)** structural representation, and **(c)** topological graph representation with the (1) rack, (2, 3) rotary bucket oil cylinder, (4) moving arm, (5) bucket, (6) connecting rod, (7) rocker, and (8, 9) lifting oil cylinder.

connected to the moving arm. It shows that the topology graph has at least two binary chains of length 2.

3. There must be a binary link as the output bucket, where one end is connected with the moving arm but cannot be directly connected with the rack.

4. There must be a moving arm, for which the one end is connected to the rack, and the other end is connected to the output bucket. One end of the lifting cylinder and the rocker should be connected with the moving arm by considering the coordination and stiffness of the mechanism.

The direct connection between the binary link and hydraulic cylinder should be avoided to overcome the excessive forces on the local components of the transmission. These excessive forces can create the imbalance. The topology graphs that fulfill the above constraint criteria are shown in the first column of Table 5.

Step 3. Specify the topology graphs that meet the structural and functional requirements of the mechanism. For example, see the specialization of Fig. 7c.

First, determine the rack. Both vertex 1 and vertex 7 are three degree vertices and can be used as racks. Vertices 1 and 7 are similar, according to the loop tree matrix, and choosing vertex 1 or vertex 7 as racks is equivalent. In this paper, vertex 1 is chosen as a rack.

Second, the original moving parts are determined. The original moving parts are binary chains of length 2 in topology graph. The edges are assigned to the two that attached to the rack. Binary chain 23 and 56 are no longer similar, according to the loop tree matrix. The binary chains 89 and 23, 89 and 56, and 56 and 23 can be chosen as the original moving parts.

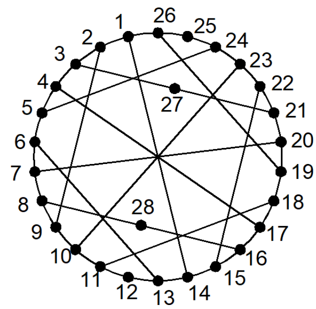


Figure 8. A 28-link PSKC.

When binary chains 89 and 23 are chosen as the original moving parts, only vertex 5 can be chosen as the output bucket. The final functional schematic is shown in Table 5.

When binary chains 56 and 23 are chosen as the original moving parts, only vertex 5 can be chosen as the output bucket. The final functional schematic is shown in Fig. 7a.

When binary chains 56 and 89 are chosen as the original moving parts, it does not satisfy the structural constraint (4).

According to the above specialization process, other topology graphs are given specifically, as shown in the second column of Table 5. In specialized topological graph, ▲ represents the rack, ○ represents output the bucket, ■ represents the moving arm, and the solid lines represent prismatic pair.

Step 4. Draw the schematic graph and delete the existing design as the rest forms the new designs, as shown in the third column of Table 5.

5 Recognition result analysis

At present, there are relatively few studies on the similarity, and the recognition methods are difficult when attempting to meet the requirements in reliability and efficiency. In this paper, a similarity recognition method based on the tree loop was proposed, which aims to provide some theoretical support for configuration synthesis and innovation.

5.1 Computational complexity

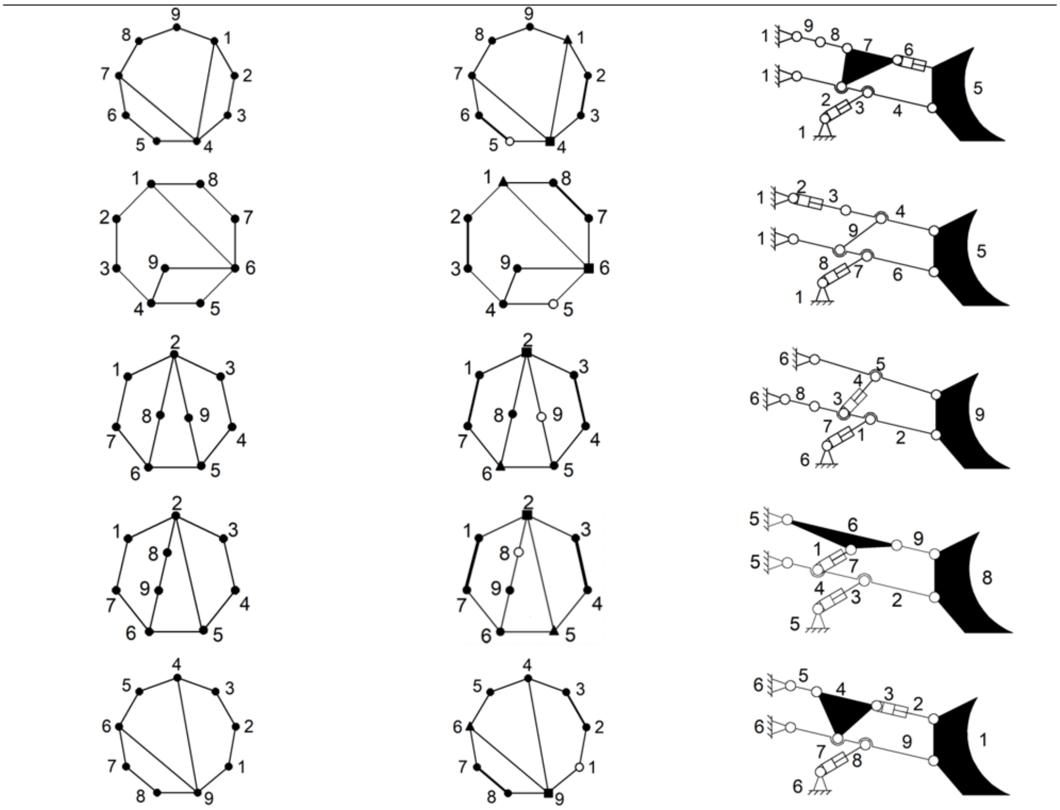
Compared with the graph theory definition method for recognizing similar vertices, the application of the LT method can reduce the computer computation considerably. Figure 8 shows the available methods for identifying similar vertices, including the definition method needs a calculation of 28! cycle times. The method of Dharanipragada and Chintada (2016) requires a calculation of $16! \times 8! \times 4!$ cycle times, and the LT method only needs $\sum_{1}^{16} 1 + \sum_{1}^{8} 1 + \sum_{1}^{4} 1$ cycle times.

5.2 Computational complexity

LTM contains almost all information of the LT, such as the vertex degree information, assignment information of

Table 5. The new designs of 6-link loader.

Topological graph	Specialization	Particularization



connecting edges, and loop information. Therefore, the LT method used to recognize the similar vertices is effective, as proven by numerous recognition examples.

6 Conclusions

In this paper, the concept of a loop tree containing multidimensional topological information is proposed, and the loop tree matrix is established for similarity recognition. This method is applied to planar single-joint KCs and contracted graphs to reflect its applicability, and its accuracy is verified by the definition method. At the same time, the efficiency of the method is proved by qualitatively comparing the calculation cycle times of various similarity recognition methods. Finally, through the application of the similarity recognition in mechanism creation, the practical significance of this method is shown, which provides a reference for mechanical equipment design.

Appendix A

Table A1. Similarity of the 10-link with 1 DOF PSKCs.

Number	Adjacency matrix	LTM method	Definition method
1	101000001100010001000100100000100001000100101	[1, 4, 7, 8] [2, 3] [5, 6, 9, 10]	Same
2	101000001100100001000010100000100001000100101	[1, 4] [2, 3] [5, 10] [6, 9] [7, 8]	
3	101000001100100001000100100000100001000100101	No	
4	100100001100100001000100100000100001000100101	No	
5	100100001100010001000100100000100001000100101	[3, 7] [4, 6]	
6	100100001100010001000000100100100001000100101	[1, 8] [2, 4, 5, 7] [3, 6] [9, 10]	
7	100000011100100001000100100001100001000100100	[1, 4, 6, 8] [2, 3] [5, 7, 9, 10]	
8	101000010100100001000001100000100001000100110	[1, 3] [9, 10]	
9	100100001100010001000010100000100001000100101	[1, 3] [4, 10] [5, 9] [6, 8]	
10	100010001100010001000000100010100001000100101	[1, 2, 6, 7] [3, 5, 8, 10] [4, 9]	
11	100100001100010001000000100010100001000100101	[1, 5] [2, 4] [6, 10] [7, 9]	
12	100100010100000011001000100001100001000100100	No	
13	100010001101000001000100100000100001000100101	No	
14	100010001101000001000010100000100001000100101	No	
15	100001001101000001000100100000100001000100101	No	
16	101000010100010001000001100000100011000100100	No	
17	101000001100001001010000100000100001000100101	No	
18	100100001100000101010000100000100001000100101	No	
19	100000011101000001000001101000100001000100100	[1, 7] [2, 4] [3, 5] [6, 10] [8, 9]	
20	101000001100000101010000100000100001000100101	[1, 4] [2, 3] [5, 10] [6, 9] [7, 8]	
21	100100010100010001000001100000100001001100100	[1, 7] [4, 10] [5, 6] [8, 9]	
22	100100001100001001001000100000100001000100101	No	
23	100100001100001001010000100000100001000100101	No	
24	100100001100001001000000101000100001000100101	[1, 8] [5, 7] [9, 10]	
25	101000001100010001000000100000101001000100101	[1, 8] [2, 4, 5, 7] [3, 6] [9, 10]	
26	100000011101000001000100100000100001001100100	[3, 5] [6, 8] [9, 10]	
27	100010010100000011000100100001100001000100100	No	
28	100100001100000101001000100000100001000100101	No	
29	101000010100001001000001100000100001001100100	No	
30	100000011100100001000001100000101001000100100	No	
31	100000011100001001010000100001100001000100100	[1, 4, 6, 8] [2, 3] [5, 9] [7, 10]	
32	100100010100000011010000100000100001000100110	[1, 2] [3, 5] [9, 10]	
33	101000001100010001000000100000100101000100101	No	
34	101000001100010001000000100000100001010100101	[1, 6] [2, 7] [3, 8] [4, 9] [5, 10]	
35	100100001100000101000000101000100001000100101	[1, 5] [2, 4] [6, 10] [7, 9]	
36	100010010100000011000100100000100011000100100	[1, 2] [3, 6] [4, 7] [5, 8] [9, 10]	
37	100001001100100001000000100010100001000100101	[1, 2, 6, 7] [3, 5, 8, 10] [4, 9]	
38	100010001100001001001000100000100001000100101	[1, 6] [2, 7] [3, 8] [4, 9] [5, 10]	
39	101000001100001001001000100000100001000100101	[1, 4, 7, 8] [2, 3] [5, 6, 9, 10]	
40	100000101101000001001000100000100001000100101	No	
41	100001010100000011010000100001100001000100100	[1, 7] [2, 6] [5, 10] [8, 9]	
42	100000101101000001000000101000100001000100101	[1, 8] [2, 7] [3, 6] [4, 5] [9, 10]	
43	100010010100010001000001100000100011000100100	[1, 7] [2, 6] [3, 5] [4, 10] [8, 9]	
44	100001001100001001010000100000100001000100101	[1, 8] [2, 7] [3, 6] [4, 5] [9, 10]	
45	100010010100001001000001100000100011000100100	[4, 10]	
46	100000110100010001000010100001100001001100000	[3, 7] [4, 6] [5, 10] [8, 9]	
47	101000010100001001000000100000100011000101100	[1, 2] [3, 9] [4, 8] [5, 7] [6, 10]	
48	100000101100010001010000100000100001000100101	[1, 3, 6, 8] [2, 7] [4, 5, 9, 10]	
49	100001010100100001000001100000100011000100100	[1, 7] [2, 6] [3, 5] [4, 10] [8, 9]	
50	100000110100100001000001100000100011000110000	[1, 3, 5, 7] [2, 6] [4, 8, 9, 10]	

Table A1. Continued.

Number	Adjacency matrix	LTM method	Definition method
51	101010010100000011000000100000100001000100101	[2, 4] [5, 10] [6, 9] [7, 8]	Same
52	101010001100000101000000100000100001000100101	No	
53	101001001100000101000000100000100001000100101	No	
54	100101001100000101000000100000100001000100101	No	
55	101010001100001001000000100000100001000100101	No	
56	101000011100001001000000100000100011000100100	[2, 4] [5, 8] [6, 7] [9, 10]	
57	101010001100000001000010100000100001000100101	No	
58	100100011100001001000000100000100001001100100	[2, 5] [3, 4] [6, 8] [9, 10]	
59	101001001100000001000010100000100001000100101	No	
60	101000011100010001000000100000100011000100100	No	
61	100101001100000001000100100000100001000100101	No	
62	101010001100000001000100100000100001000100101	No	
63	100100011100001001000001100000100001000100100	No	
64	100100011100001001000000100001100001000100100	No	
65	100100011100010001000001100000100001000100100	No	
66	100100011100010001000000100001100001000100100	No	
67	100010011100010001000000100001100001000100100	[2, 6] [3, 5]	
68	101000011100000001000100100000100011000100100	No	
69	100100011100000001000100100001100001000100100	No	
70	100000111100100001000010100001100001000100000	No	
71	100010011100000001001000100000100011000100100	No	
72	100100011100000001001000100001100001000100100	No	
73	100010011100000001001000100001100001000100100	No	
74	101000011100000001001000100000100011000100100	No	
75	101000011100000001001000100000100001001100100	No	
76	100000111100000001010000100010100011000100000	[2, 10] [3, 5]	
77	101010010100000011000000100000100001000100110	[2, 6] [3, 5] [7, 10]	
78	101000011100001001000000100000100001001100100	No	
79	100101001100000001000010100000100001000100101	[2, 10] [3, 9] [4, 8] [5, 7]	
80	100010011100001001000000100001100001000100100	No	
81	100000111100010001000010100000100011000100000	[3, 7] [4, 6] [8, 9]	
82	101000011100000001000100100000100001001100100	No	
83	100100011100000001000100100000100001001100100	No	
84	100010011100000001000100100001100001000100100	[4, 8] [5, 7] [9, 10]	
85	100000111100000001001000100010100011000100000	[2, 9] [3, 4] [5, 7] [8, 10]	
86	101001001100001001000000100000100001000100101	No	
87	101001001100000001000100100000100001000100101	No	
88	100101001100000001000000100100100001000100101	[2, 10] [3, 9] [4, 8] [5, 7]	
89	100010011100010001000001100000100001000100100	No	
90	100100011100000001000000101000100001001100100	No	
91	100100011100000001000000100001100001000101100	No	
92	100100110100000001010000100001100001000100001	[4, 6] [7, 10] [8, 9]	
93	100100101100010001000000100000100001000100101	No	
94	101000101100010001000000100000100001000100101	No	
95	100100101100100001000000100000100001000100101	[2, 5] [3, 4]	
96	100010101100000001001000100000100001000100101	No	
97	100001011100100001000001100000100001000100100	No	
98	101000101100000001001000100000100001000100101	No	
99	101010001100000001000000100000101001000100101	No	
100	101010010100000001000000100000100011000101100	No	
101	101001001100000001000000100000101001000100101	No	
102	100100011100100001000000100000100001000101100	[2, 5] [3, 4]	
103	101001010100000011000000100000100001001100100	[2, 4] [5, 10]	
104	101000101100100001000000100000100001000100101	No	
105	100001011100100001000000100001100001000100100	No	

Table A1. Continued.

Number	Adjacency matrix	LTM method	Definition method
106	10010010110000000010010001000000100001000100101	No	Same
107	10100101010000000010000011000000100001001100100	No	
108	10010010110000000010100001000000100001000100101	No	
109	10000101110100000010000011000000100001000100100	No	
110	10100010110000000010100001000000100001000100101	No	
111	10000101110000000010100001000000100011000100100	[2, 10] [3, 5]	
112	1000010111000000001010000100001100001000100100	No	
113	10010001110001000100000001000000100001000100110	[9, 10]	
114	10100001110001000100000001000000100001000100110	[9, 10]	
115	10010001110010000100000001000000100001000100110	[2, 5] [3, 4] [9, 10]	
116	1001000111000000001000001100100100001000100100	[2, 10]	
117	1000001111001000010000101000000100001000101000	[8, 10]	
118	10100001110000000010010001000000100001000100110	[9, 10]	
119	1001000111000000001000001101000100001000100100	[2, 10]	
120	1001001101000000001000010100001100001001100000	[2, 9]	
121	1000100111000000001000001101000100001000100100	[2, 10]	
122	10100001110010000100000001000000100001000101100	No	
123	10010001110000000010100001000000100001000101100	No	
124	1000001111010000010000101000000100001001100000	No	
125	10100001110000000010100001000000100001000101100	No	
126	10100011010000001100000001000000100001001110000	[2, 4] [5, 10] [8, 9]	
127	10100001110010000100000001000000100001000100110	[9, 10]	
128	1000001111001000010000000100010100001000101000	[8, 10]	
129	1000100111000000001000001100100100001000100100	[2, 10]	
130	10100011010000000010000011000000100001001110000	[8, 9]	
131	10010001110000000010100001000000100001000100110	[9, 10]	
132	1000001111010000010000101000000100001000101000	[8, 10]	
133	10100001110000000010100001000000100001000100110	[9, 10]	
134	1000001111000000001000010101000100011000100000	[2, 9] [5, 7] [8, 10]	
135	1000001111000000001010000100010100001000101000	[8, 10]	
136	10001010110100000100000001000000100001000100101	No	
137	10000101110100000100000001000000100001001100100	No	
138	10001001110100000100000001000000100001000101100	No	
139	100101010100000011000000100001100001000100100	[3, 10]	
140	100010110100000011000000100001100101000100000	[3, 10]	
141	10001001110100000100000001000000100001000100110	[9, 10]	
142	10000011110100000100000001000000100001010101000	[8, 10]	
143	100100110100000011000000100001100001010100000	[3, 10]	
144	100100110100000011000000100001100001000110000	[3, 10] [8, 9]	
145	100001110100000011000000100001101001010000000	[3, 10] [7, 9]	
146	100101010100000011000000100000100001000101100	[2, 5] [3, 4] [6, 10]	
147	101001010100000011000000100000100001000101100	No	
148	10001010110000000010001001000000100001000100101	[1, 8] [2, 7] [3, 6] [4, 5] [9, 10]	
149	100010110100000011000010100000100001001100000	No	
150	10010010110000000010001001000000100001000100101	No	
151	10100101010000000010000011000000100001000101100	No	
152	1000010111000000001001000100001100001000100100	No	
153	10001001110000000010010011000000100001000100100	[1, 3] [2, 10] [4, 9] [5, 8] [6, 7]	
154	10010001110001000100000001000000100001000101100	No	
155	10100001110001000100000001000000100001000101100	No	
156	1000001111001000010000101000000100001001100000	No	
157	10100001110000000010010001000000100001000101100	No	
158	10100011010010000100000001000000100001001100001	[7, 10] [8, 9]	
159	100001101101000001000000110000001001010000000100	[1, 5] [4, 10] [6, 7, 8, 9]	
160	10100011010000000010100001000000100001001100001	[1, 6] [2, 5] [3, 4] [7, 8, 9, 10]	

Table A1. Continued.

Number	Adjacency matrix	LTM method	Definition method
161	100101010100000000100000110000001000010001011100	No	Same
162	100010110100000000100000110001010000100110000 0	[1, 6] [2, 5, 9, 10] [3, 4] [7, 8]	
163	100100110100000001100000001000000100001000111000	[2, 5] [3, 4] [6, 10] [8, 9]	
164	101000110100000001100000001000000100001000111000	[8, 9]	
165	100100011100000000100010110000001000010001001 00	[1, 3] [2, 10] [4, 9] [5, 8] [6, 7]	
166	100000111100010001000000010001110000100010000 0	[9, 10]	
167	100100011100000000100100110000001000010001001 00	[2, 10]	
168	100000111100010001000000010000010011100010000 0	[9, 10]	
169	100000111100100001000000010001110000100010000 0	[9, 10]	
170	101000110100000001100000001000000100001011100000	[2, 4] [5, 10]	
171	100000111100100001000000010001010000100110000 0	[1, 6] [5, 9] [7, 8]	
172	100100011100000000100100010000001000010001011 00	[1, 7] [2, 6] [3, 5] [8, 9]	
173	101000110100000000100000110000001000010111000 00	No	
174	101001001100000000100000010010010000100010010 1	[1, 4] [2, 3] [5, 10] [6, 9] [7, 8]	
175	100100110100100001000000010000001000010011000 01	[1, 6] [2, 5] [3, 4] [7, 8, 9, 10]	
176	100100110100000000100001110000001000010001010 00	[2, 9]	
177	100000111100000000100100010000001001110001000 00	[1, 5] [2, 4, 6, 8] [3, 7] [9, 10]	
178	100101010100000001100000001000000100011000100100	No	
179	100010101100000000101000010000001000010001001 01	[1, 6] [2, 7] [3, 8] [4, 9] [5, 10]	
180	100010110100000001100000001000000100111000100000	No	
181	100001011101000000100000001000000100011000100100	No	
182	100101100100000001100000001000111000010000000100	[3, 10]	
183	100100110100000001100000001000000100011010100000	[1, 5] [2, 6] [3, 7] [4, 8] [9, 10]	
184	100101010100000000100000110000001000110001001 00	[4, 10]	
185	100010110100000001100000001000111000010001000 00	[3, 10]	
186	100000111101000000100000001000000100101000101000	[8, 10]	
187	100100110100000001100000001000000100011000110000	[8, 9]	
188	100001110100000001100000001001011001010000000000	[3, 10]	
189	100011010100000001100000001001011001000000100000	[1, 4] [2, 5] [3, 6, 9, 10] [7, 8]	
190	100010110101000000100000001000000100011000100001	[1, 5] [2, 6] [3, 7] [4, 8] [9, 10]	
191	100100110100000000100001010000001000110001010 00	[1, 5] [2, 6, 9, 10] [3, 7] [4, 8]	
192	100001110100000001100000001001011000010100000 00	[3, 10] [7, 9]	
193	101000101100000000100000010100010000100010010 1	[1, 4] [2, 3] [5, 10] [6, 9] [7, 8]	
194	100001011100000000100000110100010000100010010 0	[2, 10]	
195	100001011100000000101000110000001000010001001 00	[1, 3] [2, 10] [4, 9] [5, 8] [6, 7]	
196	100010110100000000100001010000110000100110000 0	[1, 6] [2, 5, 9, 10] [3, 4] [7, 8]	
197	100010110100000000100001110000001000110001000 00	[2, 9] [4, 10]	
198	100001110100000000100010110000001000110100000 00	[1, 3] [2, 8] [4, 7, 9, 10] [5, 6]	
199	100010110101000000100000001000000100001001100001	[1, 6] [2, 5] [3, 4] [7, 8, 9, 10]	
200	100101100100000001100000001000011001010000000100	[1, 5] [2, 4] [3, 10] [6, 7, 8, 9]	
201	100000111101000000100000001000000100001011100000	[1, 6] [2, 5] [3, 4] [7, 8] [9, 10]	
202	100001110100000001100000001000011011010000000000	[1, 5] [2, 4] [3, 10] [6, 7] [8, 9]	
203	101001010100000000100000001000011000010001011 00	[2, 9] [3, 8] [4, 7] [5, 6]	
204	100010110100000000100001110000001000010011000 00	[2, 9]	
205	100001110100000000100010110000001001110000000 00	[1, 5] [2, 4, 8, 10] [6, 7]	
206	101010101100000000100000010000001000010001001 01	[2, 10] [3, 9] [4, 8] [5, 7]	
207	100101011100000000100000010000110000100010010 0	No	
208	101001011100000000100000010000001000110001001 00	No	
209	101000111100000000100000010000001001010011000 00	No	
210	100101101100000000100000110001010000100000001 00	[2, 10]	
211	100100111100000000100000010001010000100110000 0	[2, 8] [3, 7] [4, 6] [9, 10]	
212	100101011100000000100000110000001000010001001 00	[2, 10]	
213	100010111100000000100001010000001000110001000 00	[2, 9]	
214	100010111100000000100001010000110000100010000 0	[2, 9]	
215	100100111100000000100001010000001000010011000 00	[2, 9]	

Table A1. Continued.

Number	Adjacency matrix	LTM method	Definition method
216	100001111100000000100010010001010001100000000	[2, 8]	Same
217	100011011100000000100001010010010001000010000 0	[2, 6, 9, 10] [3, 5]	
218	101010110100000000100000010000010001100010000 1	[2, 8] [3, 7] [4, 6]	
219	100100111100000000100001010000001000010001010 00	[2, 8, 9, 10] [3, 7] [4, 6]	
220	100001111100000000100010010001010000100100000 0	[2, 8] [7, 10]	
221	101010110100000000100000010000110000100010000 1	[2, 9] [3, 10]	
222	101001101100000000100000010001010001100000010 0	[2, 8] [3, 9]	
223	100101101100000000100000010001110000100000010 0	[2, 8] [3, 9]	
224	100101101100000000100000110000010010100000010 0	[2, 10] [6, 9] [7, 8]	
225	100010111100000000100000010001110000100010000 0	[9, 10]	
226	100001111100000000100000010011010001100000000 0	[8, 9]	
227	100011011100000000100001010010110000000010000 0	[2, 9] [5, 8] [6, 7]	
228	100001111100000000100010010000010011100000000 0	[2, 8] [9, 10]	
229	101011010100000000100000010010110000000010000 1	[1, 4] [2, 3, 5, 6, 7, 8, 9, 10]	
230	100011011100000000100000010011110000000010000 0	[1, 4] [2, 3, 5, 6, 7, 8] [9, 10]	

Appendix B

Table B1. L_d of a 10-link with 1 DOF PSKCs.

Adjacency matrix	L_d of each vertex	Similarity
$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	(1) $\begin{bmatrix} 331 & 321 & 321 & 321 & 311 & 311 & 211 & 211 & 3 & 0 & 0 \\ 331 & 321 & 321 & 321 & 311 & 311 & 3 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 321 & 311 & 311 & 3 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 321 & 311 & 211 & 211 & 311 & 3 & 0 & 0 \\ 331 & 321 & 321 & 321 & 211 & 211 & 311 & 311 & 211 & 211 & 3 \\ 331 & 321 & 0 & 321 & 211 & 211 & 3 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 321 & 211 & 211 & 3 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 321 & 211 & 211 & 3 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 321 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 321 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 331 & 321 & 321 & 211 & 211 & 321 & 311 & 311 & 211 & 211 & 3 \\ 331 & 321 & 321 & 211 & 211 & 321 & 3 & 0 & 0 & 0 & 0 \\ 331 & 321 & 211 & 211 & 321 & 321 & 311 & 311 & 211 & 211 & 3 \\ 331 & 321 & 211 & 211 & 321 & 321 & 311 & 311 & 3 & 0 & 0 \\ 331 & 321 & 211 & 211 & 321 & 321 & 211 & 211 & 3 & 0 & 0 \\ 331 & 321 & 211 & 211 & 321 & 321 & 3 & 0 & 0 & 0 & 0 \\ 331 & 211 & 211 & 321 & 321 & 321 & 311 & 311 & 3 & 0 & 0 \\ 331 & 211 & 211 & 321 & 321 & 321 & 311 & 211 & 211 & 311 & 3 \\ 331 & 211 & 211 & 321 & 321 & 321 & 211 & 211 & 311 & 311 & 3 \\ 331 & 211 & 211 & 321 & 321 & 321 & 3 & 0 & 0 & 0 & 0 \\ 331 & 211 & 211 & 321 & 321 & 321 & 3 & 0 & 0 & 0 & 0 \\ 331 & 211 & 211 & 321 & 321 & 321 & 3 & 0 & 0 & 0 & 0 \\ 331 & 211 & 211 & 321 & 321 & 211 & 211 & 321 & 311 & 311 & 3 \\ 331 & 211 & 211 & 321 & 321 & 211 & 211 & 321 & 3 & 0 & 0 \end{bmatrix}$	[1, 4, 7, 8] [2, 3] [5, 6, 9, 10]

Table B1. Continued.

Adjacency matrix	L_d of each vertex												Similarity
	(2)	331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	211	211	311	311	3	0	0	
		331	321	321	321	211	211	311	311	3	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	211	211	321	311	311	3	0	0	
		331	321	321	211	211	321	311	311	3	0	0	
		331	321	321	211	211	321	311	211	211	311	3	
		331	321	321	211	211	321	311	211	211	311	3	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	211	211	321	321	311	211	211	311	3	
		331	321	211	211	321	321	311	211	211	311	3	
		331	321	211	211	321	321	211	211	311	311	3	
		331	321	211	211	321	321	211	211	311	311	3	
		331	321	211	211	321	321	3	0	0	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
	(3)	331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	211	211	311	311	3	0	0	
		331	321	321	321	211	211	311	311	3	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	211	211	321	311	311	3	0	0	
		331	321	321	211	211	321	311	311	3	0	0	
		331	321	321	211	211	321	311	211	211	311	3	
		331	321	321	211	211	321	311	211	211	311	3	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	211	211	321	321	311	211	211	311	3	
		331	321	211	211	321	321	311	211	211	311	3	
		331	321	211	211	321	321	211	211	311	311	3	
		331	321	211	211	321	321	211	211	311	311	3	
		331	321	211	211	321	321	3	0	0	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	321	211	211	321	321	3	0	0	0	0	

Table B1. Continued.

Adjacency matrix	L_d of each vertex												Similarity
	(4)	331	321	321	321	311	311	211	211	3	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	211	211	311	3	0	0	
		331	321	321	321	211	211	311	311	211	211	3	
		331	321	0	321	211	211	3	0	0	0	0	
		331	321	321	321	211	211	3	0	0	0	0	
		331	321	321	321	211	211	3	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	211	211	321	311	311	211	211	3	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	211	211	321	321	311	311	211	211	3	
		331	321	211	211	321	321	311	311	3	0	0	
		331	321	211	211	321	321	211	211	3	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	211	211	321	321	321	311	311	3	0	0	
		331	211	211	321	321	321	311	211	211	311	3	
		331	211	211	321	321	321	211	211	311	311	3	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	211	211	321	311	311	3	
		331	211	211	321	321	211	211	321	3	0	0	
	(5)	221	321	321	321	311	311	311	211	2	0	0	
		221	321	321	321	311	211	211	311	311	211	2	
		221	321	321	321	311	211	2	0	0	0	0	
		221	321	321	321	311	211	2	0	0	0	0	
		221	321	321	321	311	211	2	0	0	0	0	
		221	321	321	321	211	211	311	311	311	211	2	
		221	321	321	211	211	321	311	311	311	211	2	
		221	321	321	211	211	321	311	211	2	0	0	
		221	211	321	321	321	311	311	311	2	0	0	
		221	211	321	321	321	311	211	211	311	311	2	
		221	211	321	321	321	311	2	0	0	0	0	
		221	211	321	321	321	311	2	0	0	0	0	
		221	211	321	321	321	311	2	0	0	0	0	
		221	211	321	321	321	211	211	311	311	311	2	
		221	211	321	321	211	211	321	311	311	311	2	
		221	211	321	321	211	211	321	311	2	0	0	
	(6)	221	321	321	321	311	311	311	211	2	0	0	
		221	321	321	321	311	211	211	311	311	211	2	
		221	321	321	321	311	211	2	0	0	0	0	
		221	321	321	321	311	211	2	0	0	0	0	
		221	321	321	321	311	211	2	0	0	0	0	
		221	321	321	321	211	211	311	311	311	211	2	
		221	321	321	211	211	321	311	311	311	211	2	
		221	321	321	211	211	321	311	211	2	0	0	
		221	211	321	321	321	311	311	311	2	0	0	
		221	211	321	321	321	311	211	211	311	311	2	
		221	211	321	321	321	311	2	0	0	0	0	
		221	211	321	321	321	311	2	0	0	0	0	
		221	211	321	321	321	311	2	0	0	0	0	
		221	211	321	321	321	211	211	311	311	311	2	
		221	211	321	321	211	211	321	311	311	311	2	
		221	211	321	321	211	211	321	311	2	0	0	

Table B1. Continued.

Adjacency matrix	L_d of each vertex												Similarity
	(7)	331	321	321	321	311	311	211	211	3	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	211	211	311	3	0	0	
		331	321	321	321	211	211	311	311	211	211	3	
		331	321	0	321	211	211	3	0	0	0	0	
		331	321	321	321	211	211	3	0	0	0	0	
		331	321	321	321	211	211	3	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	211	211	321	311	311	211	211	3	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	211	211	321	321	311	311	211	211	3	
		331	321	211	211	321	321	311	311	3	0	0	
		331	321	211	211	321	321	211	211	3	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	211	211	321	321	321	311	311	3	0	0	
		331	211	211	321	321	321	311	211	211	311	3	
		331	211	211	321	321	321	211	211	311	311	3	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	211	211	321	311	311	3	
		331	211	211	321	321	211	211	321	3	0	0	
	(8)	331	321	321	321	311	311	211	211	3	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	311	3	0	0	0	0	
		331	321	321	321	311	211	211	311	3	0	0	
		331	321	321	321	211	211	311	311	211	211	3	
		331	321	0	321	211	211	3	0	0	0	0	
		331	321	321	321	211	211	3	0	0	0	0	
		331	321	321	321	211	211	3	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	321	3	0	0	0	0	0	0	
		331	321	321	211	211	321	311	311	211	211	3	
		331	321	321	211	211	321	3	0	0	0	0	
		331	321	211	211	321	321	311	311	211	211	3	
		331	321	211	211	321	321	311	311	3	0	0	
		331	321	211	211	321	321	211	211	3	0	0	
		331	321	211	211	321	321	3	0	0	0	0	
		331	211	211	321	321	321	311	311	3	0	0	
		331	211	211	321	321	321	311	211	211	311	3	
		331	211	211	321	321	321	211	211	311	311	3	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	321	3	0	0	0	0	
		331	211	211	321	321	211	211	321	311	311	3	
		331	211	211	321	321	211	211	321	3	0	0	

Table B1. Continued.

Adjacency matrix	L_d of each vertex	Similarity
	(9) $\begin{bmatrix} 221 & 321 & 321 & 321 & 311 & 311 & 311 & 211 & 2 & 0 & 0 \\ 221 & 321 & 321 & 321 & 311 & 211 & 211 & 311 & 311 & 211 & 2 \\ 221 & 321 & 321 & 321 & 311 & 211 & 2 & 0 & 0 & 0 & 0 \\ 221 & 321 & 321 & 321 & 311 & 211 & 2 & 0 & 0 & 0 & 0 \\ 221 & 321 & 321 & 321 & 311 & 211 & 2 & 0 & 0 & 0 & 0 \\ 221 & 321 & 321 & 321 & 211 & 211 & 311 & 311 & 311 & 211 & 2 \\ 221 & 321 & 321 & 211 & 211 & 321 & 311 & 311 & 311 & 211 & 2 \\ 221 & 321 & 321 & 211 & 211 & 321 & 311 & 211 & 2 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 311 & 311 & 2 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 211 & 211 & 311 & 311 & 2 \\ 221 & 211 & 321 & 321 & 321 & 311 & 2 & 0 & 0 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 2 & 0 & 0 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 2 & 0 & 0 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 211 & 211 & 311 & 311 & 311 & 2 \\ 221 & 211 & 321 & 321 & 211 & 211 & 321 & 311 & 311 & 311 & 2 \\ 221 & 211 & 321 & 321 & 211 & 211 & 321 & 311 & 2 & 0 & 0 \end{bmatrix}$	
	(10) $\begin{bmatrix} 221 & 321 & 321 & 321 & 311 & 311 & 311 & 211 & 2 & 0 & 0 \\ 221 & 321 & 321 & 321 & 311 & 211 & 211 & 311 & 311 & 211 & 2 \\ 221 & 321 & 321 & 321 & 311 & 211 & 2 & 0 & 0 & 0 & 0 \\ 221 & 321 & 321 & 321 & 311 & 211 & 2 & 0 & 0 & 0 & 0 \\ 221 & 321 & 321 & 321 & 311 & 211 & 2 & 0 & 0 & 0 & 0 \\ 221 & 321 & 321 & 321 & 211 & 211 & 311 & 311 & 311 & 211 & 2 \\ 221 & 321 & 321 & 211 & 211 & 321 & 311 & 311 & 311 & 211 & 2 \\ 221 & 321 & 321 & 211 & 211 & 321 & 311 & 211 & 2 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 311 & 311 & 2 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 211 & 211 & 311 & 311 & 2 \\ 221 & 211 & 321 & 321 & 321 & 311 & 2 & 0 & 0 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 2 & 0 & 0 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 311 & 2 & 0 & 0 & 0 & 0 \\ 221 & 211 & 321 & 321 & 321 & 211 & 211 & 311 & 311 & 311 & 2 \\ 221 & 211 & 321 & 321 & 211 & 211 & 321 & 311 & 311 & 311 & 2 \\ 221 & 211 & 321 & 321 & 211 & 211 & 321 & 311 & 2 & 0 & 0 \end{bmatrix}$	

Code and data availability. Code and data can be made available upon request.

Author contributions. LW was responsible for the conceptualization, methodology, and preparing the original draft. RC and YX curated the data, reviewed and edited the paper, developing the software, and validating the research. LS, GY, and CW took responsibility for supervising and the project administration. All authors have read and agreed to the published version of the paper.

Competing interests. The contact author has declared that neither they nor their co-authors have any competing interests.

Financial support. This work has been supported in part by the National Natural Science Foundation of China (grant no. 51975534), the Zhejiang Provincial Key Research and Development Program (grant no. 2022C02006), the high-level talents research start-up project of the Hangzhou Vocational and Technical College (grant no. HZYGCC202103), and the Public Welfare Technology Application Research Project of Zhejiang Province (grant no. LGN21E050001).

Review statement. This paper was edited by Daniel Condurache and reviewed by two anonymous referees.

References

Ambekar, A. G. and Agrawal, V. P.: Canonical numbering of kinematic chains and isomorphism problem: min code, Mech. Mach. Theory, 22, 453–461, 1987.

Cubillo, J. P. and Wan, J.: Comments on mechanism kinematic chain isomorphism identification using adjacent matrices, Mech. Mach. Theory, 40, 131–139, 2005.

Dharanipragada, V., and Chintada, M.: Split Hamming String as an Isomorphism Test for One Degree-of-Freedom Planar Simple-Jointed Kinematic Chains Containing Sliders, J. Mech. Design, 138, 082301, https://doi.org/10.1115/1.4033611, 2016.

Freudenstein, F.: The basic concepts of Polya's theory of enumeration, with application to the structural classification of mechanisms, J. Mechanisms, 2, 275–290, 1967.

Harary, F. and Palmer, E. M.: On similar points of a graph, J. Math. Mech., 15, 623–630, 1966.

He, P. R., Zhang, W. J., and Li, Q.: Some further development on the eigensystem approach for graph isomorphism detection, J. Frankl. Inst., 342, 657–673, 2005.

Lv, T.: Sufficient and Necessary Conditions for Similarity, J. Beijing Univ. Technol., 15, 63–66, 1989.

Mruthyunjaya, T. S.: A computerized methodology for structural synthesis of kinematic chains: Part 1 – Formulation, Mech. Mach. Theory, 19, 487–495, 1984.

Mruthyunjaya, T. S. and Balasubramanian, H. R.: In quest of a reliable and efficient computational test for detection of isomorphism in kinematic chains, Mech. Mach. Theory, 22, 131–139, 1987.

Rai, R. K. and Punjabi, S.: Kinematic chains isomorphism identification using link connectivity number and entropy neglecting tolerance and clearance, Mech. Mach. Theory, 123, 40–65, 2018.

Rai, R. K. and Punjabi, S.: A new algorithm of links labelling for the isomorphism detection of various kinematic chains using binary code, Mech. Mach. Theory, 131, 1–32, 2019.

Rao, A. C. and Rao, C. N.: Loop based pseudo hamming values – I testing isomorphism and rating kinematic chains, Mech. Mach. Theory, 28, 113–127, 1993.

Rao, A. C. and Varada Raju, D.: Application of the hamming number technique to detect isomorphism among kinematic chains and inversions, Mech. Mach. Theory, 26, 55–75, 1991.

Rao, Y. V. D. and Rao, A. C.: Generation of Epicyclic Gear Trains of One Degree of Freedom, J. Mech. Design, 130, 232–245, 2008.

Shin, J. K. and Krishnamurty, S.: Development of a standard code for colored graphs and its application to kinematic chains, International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, 9419, 247–254, 1992.

Sun, L., Cui, R., Ye, Z., Zhou, Y., and Wu, C.: Similarity recognition and isomorphism identification of planar kinematic chains, Mech. Mach. Theory, 145, 103678, https://doi.org/10.1016/j.mechmachtheory.2019.103678, 2020.

Sun, L., Ye, Z., Cui, R., Huang, X., and Wu, C.: Eliminating isomorphism identification method for synthesizing nonfractionated kinematic chains based on graph similarity, Mech. Mach. Theory, 167, 104500, https://doi.org/10.1016/j.mechmachtheory.2021.104500, 2022.

Sun, W., Kong, J., and Sun, L.: The improved hamming number method to detect isomorphism for kinematic chain with multiple joints, J. Adv. Mech. Des. Syst., 11, JAMDSM0061, https://doi.org/10.1299/jamdsm.2017jamdsm0061, 2017.

Sunkari, R. P. and Schmidt, L. C.: Reliability and Efficiency of the Existing Spectral Methods for Isomorphism Detection, J. Mech. Design, 128, 1246–1252, 2006.

Tang, C. and Liu, T.: The Degree Code – A New Mechanism Identifier, J. Mech. Design, 115, 627–630, 1993.

Tsai, L. W.: Mechanism Design: Enumeration of Kinematic Structures According to Function, Appl. Mech. Rev., 122, B85–B86, 2000.

Uicker Jr., J. J. and Raicu, A.: A method for the identification and recognition of equivalence of kinematic chains, Mech. Mach. Theory, 10, 375–383, 1975.

Venkata Kamesh, V., Mallikarjuna Rao, K., and Annambhotla, B. S. R.: An Innovative Approach to Detect Isomorphism in Planar and Geared Kinematic Chains Using Graph Theory, J. Mech. Design, 139, 122301, https://doi.org/10.1115/1.4037628, 2017.

Wang, Y. X. and Yan, H. S.: Computerized rules-based regeneration method for conceptual design of mechanisms, Mech. Mach. Theory, 37, 833–849, 2002.

Yadav, J. N., Pratap, C. R., and Agrawal, V. P.: Computer aided detection of isomorphism among binary chains using the link-link multiplicity distance concept, Mech. Mach. Theory, 31, 873–877, https://doi.org/10.1016/0094-114X(96)00002-X, 1996.

Yan, H. S.: A methodology for creative mechanism design, Mech. Mach. Theory, 27, 235–242, 1992.

Yan, H. S. and Hall, A. S.: Linkage Characteristic Polynomials: Assembly Theorems, Uniqueness, J. Mech. Design, 104, 11–20, 1982.