Using Rows and Columns of Distance Matrix to Identify Isomorphisms in Kinematic Chains

Mohamed Aly Abdel Kader* and Abdeslam Aannaque

Mechanical Department, Mohammadia School of Engineering, Rabat, Morocco
Email: abdeslam.aannaque@emi.um5.ac.ma (A.A.)
*Correspondence: mohamedaly_abdelkader@um5.ac.ma (M.A.A.K.)

Abstract—There has always been a need to develop simple, reliable, and efficient methods for identifying isomorphic kinematic chains (KCs). Discriminating against a large number of KCs in a short period of time is a complex and difficult task at the moment. Most isomorphism identification techniques involve complex concepts and intermediate parameter comparisons, especially as the number of bars increases. The proposed method identifies isomorphism in KCs by generating an invariant from the rows and columns of the distance matrix. All of the results obtained using this method on 8-bar, 10-bar, and 12-bar, three complex 13-bar, 15-bar, and 28-bar simple joint planar kinematic chains, as well as 10-bar and 12-bar simple joint non-planar kinematic chains, agree with the published results. The method's reliability and efficiency are confirmed when the results are compared to previously published works.

Keywords—adjacency matrix, isomorphism, distance matrix, invariant identification

I. Introduction

Many researchers have been working for a long time to solve the core challenge of kinematic chains by identifying isomorphisms in a short amount of time using efficient, robust, and simple approaches. Before arriving at a satisfactory solution, the mechanical designer must consider numerous potential solutions. He should focus on solutions that are not equivalents or isomorphic to save time. As a result, identifying isomorphisms in KCs can reduce the difficulty and complexity of solving these types of problems, making them more manageable. As the number of bars increases, so does the complexity of isomorphism detection. This is due to the intermediate parameter calculations and comparisons required before reaching a type of invariant that allows for isomorphism identification. However, detecting isomorphism in a short period of time using an invariant, such as a distance matrix, presents a practical and acceptable method of improving KC design efficiency. The following is a summary of various works on the subject published in recent years by researchers. Varadaraju et al. [1] proposed a method for separating the adjacent matrix into three components using the hamming number. Vinjiamuri et al. [2] proposed using the distance between distinct vertices in kinematic graphs to calculate the called net distance to detect isomorphism. Wenjian Yang et al. [3] identified isomorphism among planetary gear trains using the perimeter loop approach. Nonetheless, in some cases, their method was unable to detect isomorphisms. Leiying et al. [4] proposed using the high-order adjacent link value to determine isomorphisms in kinematic chains. Ankur et al. [5] described an index for detecting isomorphism based on the distance matrix and the links degree matrix. Sun [6] proposed extracting joint and link codes from a joint-joint matrix to generate joint and link attributes for identifying isomorphic KCs. Using eigenvalues and eigenvectors, Chang et al. [7] proposed and mathematically demonstrated a method for identifying isomorphic kinematic chains. Yang et al. [8] proposed a method for constructing a new matrix G from the incidence matrix and then computing its row sums to generate a column vector SGf. The determinant indicates the mechanism configuration for matrix G, as well as the number and value of elements in matrix SGf. Mahmoud et al. [9] proposed generating a new concept of joint degree for all joints in a given KC from joint configuration and a unified loop array for establishing a unified chain matrix to identify isomorphism among KCs. Ding et al. [10] proposed a kinematic chain relabeling method that begins by searching for a KC configuration's perimeter loop, then relabeling the configuration based on the perimeter loop to establish the adjacent matrix corresponding to the newly relabeled KC, and finally generating code from the adjacent matrix's up triangle to identify isomorphism. Rai and Punjabi [11] presented a second relabeling method that identified isomorphism by labeling the vertices with binary coding. Yu et al. [12] used the link-assortment adjacency matrix and the binary link path to transform the high-ranking adjacency matrix to a low-ranking adjacency matrix in order to detect isomorphisms in KCs. Rongjiang Cui et al. [13] synthesized Planar KCs with P-Pairs by using the power of the adjacency matrix and the shortest distance between the links. Sun et al. [14] developed a technique for eliminating isomorphism in contract graphs after the addition of a binary vertex. Sun et al. [15] proposed using a joint matrix and improving the hamming matrix to identify isomorphisms in KCs with multiple joints. Eleashy [16] proposed combining the joint

Manuscript received September 1, 2022; revised October 20, 2022; accepted January 31, 2023.

doi: 10.18178/ijmerr.12.3.159-168

identification code (JIC) and the joint sorting code to obtain the KCSM (JSC). The rows of the KCSM allow for the identification of isomorphisms between 8-link kinematic chains with up to three prismatic pairs. This method, however, requires first determining the cycle basis in the kinematic chain graph. Similarly, Yang et al. [17] proposed a fully automatic method for the structural synthesis of planar kinematic chains with prismatic pairs that generates code that can be used to eliminate isomorphism obtained by grouping and re-labelling vertices using newly proposed degree, weight, degree sequence, and weight sequence. Deng et al. [18] used molecular topology to detect isomorphisms in topological kinematic chains, starting with the observation that the chemical molecular model is similar to a kinematic chain. Sun et al. [19] developed a method for identifying isomorphisms in kinematic chains by constructing a code identification of each vertex based on the cubic power of the adjacency matrix. Rai and Punjabi [20] described an entropy-based method for finding two invariants (while ignoring link tolerance and joint clearance) for detecting isomorphism in kinematic chains. Qing et al. [21] published a simple method for numbering the vertices by assigning an identity code to each vertex in order to detect similar vertices between each pair of KCs. Rizvi et al. [22] proposed using the adjacency matrix to generate a unique chain identification number. This identifying number was used to identify isomorphisms. He et al. [23] proposed a method for detecting isomorphism based on permutation operations to calculate eigenvectors and eigenvalues of adjacency matrices (with modifications to the previous methods). Ding and Huang [24] used labelled vertex coordinates on perimeter topological networks to generate the characteristic adjacency matrix of a canonical perimeter topological graph. They then obtained characteristic representation code, which was used to create a program that automatically draws topological graphs of kinematic chains. Ding and Huang [25] proposed generating two fundamental loop operations using the array representation of loops in topological graphs of kinematic chains to determine isomorphism. After making some changes, Ding and Huang [26, 27] improved their work from [24] to identify isomorphisms in kinematic chains. Most of the methods mentioned above are difficult for general readers to use because they involve long and complicated mathematical concepts, though each method has benefits and drawbacks. As a result, the simple method presented in this article is based on solving the isomorphism identification objective using a distance matrix. The development of new kinematic chains is central to the design of mechanisms and machines. The duplication of kinematic chains caused by isomorphisms, on the other hand, is one of the most difficult challenges for designers in this field. As the number of bars increases, the problem becomes more complicated. This problem was solved with the help of an efficient method presented in this study, which significantly reduces computation time. This method can also be used to identify isomorphisms in other fields such as chemistry and biological networks.

This paper is organized as follows: Section II explains the basic concept of the proposed method. Section III provides three examples to demonstrate the effectiveness of the proposed method. Section IV contains comparative analyses using various identification methods. The conclusion is given in Section V.

II. PROPOSED METHOD

This method involves defining the shortest path between each pair of vertices in the mechanism configuration in a distance matrix DM, then sorting each row of the distance matrix in ascending (or descending) order and repeating the process for each column of the same matrix (SDM). If two kinematic chains have different sort distance matrices, they are not isomorphic; otherwise, if the two distance matrix are identic, extract the columns corresponding to binary vertices (CDX) and similarly extract the rows corresponding to non-binary vertices (RDX) from the distance matrix. Using an injective function, multiply these two extracted matrices to compute the image of CDX and RDX. The image is calculated as follows: Consider the first row of RDX and the first column of CDX. Calculate $f(a_i,b_i)$ for each pair (a_i,b_i), where a_i is ith element of the first row of RDX and b_i the ith element of the first column of CDX. The injective function f will be discussed further below. The result is a vector IV with n elements, where n is the number of vertices. Then sort IV in descending order. Finally, append all of the sorted IV's elements to form an integer, then divide its value by 10^(2(n-2)). Repeat this process for each RDX row and CDX column combination. Enter the calculated number for each combination into a nb by nnb matrix (MI), where nb is the number of binary bars and nnb is the number of non-binary bars. The resulting matrix is then sorted by rows and columns to produce (SMI) an invariant that can be used to quickly find isomorphisms in KCs.

A function is injective if for any a and b of a domain X, if f(a) = f(b) then a = b. Equivalently, a ≠ b implies f(a) ≠ f(b).

The injective function used in this article is from N^2 to N and is defined as follows:

$$f(x, y) \to x + \frac{(x+y)(x+y+1)}{2}$$

The Distance Matrix (DM) in this approach has the advantage of being a characteristic constant related to the topology of the kinematic chain graph. As shown in Figs 5 and 6, some graphs in the literature have identical sort distance matrices but are not isomorphic. In such cases, the product of the extracted matrices from the distance matrix is required to generate a unique identity matrix for each kinematic chain. Two kinematic chains are isomorphic if every vertex in one of them corresponds to a vertex in the other chain with the same degree and exact distances from other vertices, which is required to produce an identical identity matrix. Because the identity matrix is unique, this method works for any number of vertices, especially a large number of vertices.

The distance matrix determines the shortest path between two KC vertices. It is obtained by modifying the

Floyd-Warshall method, which was originally designed for distance matrices.

Mathematical Proof

Two matrices with different sorted distance matrices are obviously not isomorphic. However, if the two sorted distance matrices are identical, the identity matrix ensures that the corresponding KCs are identified. Because the IV elements are calculated by an injective function, the antecedents for each image are guaranteed to be unique, and dividing by 10^(2(n-2)) does not change the uniqueness because all kinematic chains with the same number of bars are divided by the same number. In other words, if two kinematic chains have the same SMI, they are isomorphic; otherwise, they are not.

Example Application of this process on kinematic chain of Fig. 1

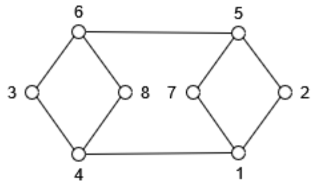


Figure 1. Eight bars KC

Distance matrix

$$\boldsymbol{DM} = \begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 1 & 2 \\ 1 & 0 & 3 & 2 & 1 & 2 & 2 & 3 \\ 2 & 3 & 0 & 1 & 2 & 1 & 3 & 2 \\ 1 & 2 & 1 & 0 & 3 & 2 & 2 & 1 \\ 2 & 1 & 2 & 3 & 0 & 1 & 1 & 2 \\ 3 & 2 & 1 & 2 & 1 & 0 & 2 & 1 \\ 1 & 2 & 3 & 2 & 1 & 2 & 0 & 3 \\ 2 & 3 & 2 & 1 & 2 & 1 & 3 & 0 \end{bmatrix} \quad (1)$$

The non-binary vertices in Fig. 1 are 1, 4, 5, and 6. The RDX represents the rows of the DM that correspond to these non-binary vertices.

$$\boldsymbol{RDX} = \boldsymbol{DM}([\boldsymbol{1},\boldsymbol{4},\boldsymbol{5},\boldsymbol{6}], :) = \begin{bmatrix} 0 & 1 & 2 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 1 & 0 & 3 & 2 & 2 & 1 \\ 2 & 1 & 2 & 3 & 0 & 1 & 1 & 2 \\ 3 & 2 & 1 & 2 & 1 & 0 & 2 & 1 \end{bmatrix} \quad (2)$$

The binary vertices are 2, 3, 7, and 8, as shown in Fig 1. The CDX is made up of the DM columns that correspond to the binary vertices.

$$\boldsymbol{CDX} = \boldsymbol{DM}(:, [\boldsymbol{2},\boldsymbol{3},\boldsymbol{7},\boldsymbol{8}]) = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 0 & 3 & 2 & 3 \\ 3 & 0 & 3 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 2 & 3 & 0 & 3 \\ 3 & 2 & 3 & 0 \end{bmatrix} \quad (3)$$

Using the injective function, compute the image of the first row of RDX and the first column of CDX.

$$RDX(1,:) = [0\ 1\ 2\ 1\ 2\ 3\ 1\ 2]$$

$$CDX(:,1) = \begin{bmatrix} 1 \\ 0 \\ 3 \\ 2 \\ 1 \\ 2 \\ 2 \\ 3 \end{bmatrix}$$

$$IV = [f(0,1)\ f(1,0)\ f(2,3)\ f(1,2) \ldots]$$
$$IV = [5\ 12\ 11\ 11\ 3\ 12\ 13\ 4]$$

Sort IV in descending order

$$IV = [13\ 12\ 12\ 11\ 11\ 5\ 4\ 3]$$

Then, arrange the elements of IV to form a single number and divide by 10^(2×(n-2)) = 10^12

$$m_{11} = IV = \frac{1312121111543}{10^{12}} = 1.312121111543$$

This value corresponds to $M_I(1,1)$. Similarly, calculate m_{12} as the image of the first row of RDX and the second column of CDX and so on.

The identity matrix obtained after dealing with all rows of RDX and all columns of CDX is:

$$\boldsymbol{M_I} = \begin{bmatrix} 1.3121 & 1.3121 & 0.0182 & 0.0182 \\ 1.3121 & 1.3121 & 0.0182 & 0.0182 \\ 0.0182 & 0.0182 & 1.3121 & 1.3121 \\ 0.0182 & 0.0182 & 1.3121 & 1.3121 \end{bmatrix} \quad (4)$$

The outcome matrix is a unique identity matrix that defines the KC. To identify isomorphisms, we simply sort this matrix and compare it to other identity matrices, as shown in the examples below.

III. APPLICATION OF THE PROPOSED METHOD

A. Application on Ten Vertices Isomorphism Identification

The vertices in Figs. 2 and 3 are clearly similar, and a simple permutation between vertices 2 and 8 results in the same configuration.

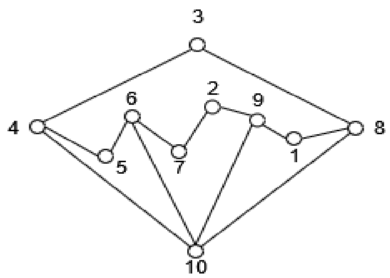


Figure 2. Ten bar kinematic chain

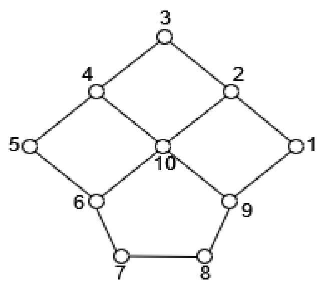


Figure 3. Ten bar KC isomorph to that of Fig.2

Because the sorted distance matrices SDM_2 and SDM_3 are identical, the extracted matrices for the graphs in Figs 2 and 3 are:

$$\boldsymbol{RD3} = \begin{bmatrix} 1 & 0 & 1 & 2 & 3 & 2 & 3 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 & 3 & 2 & 1 \\ 3 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 2 & 1 \\ 1 & 2 & 3 & 2 & 3 & 2 & 2 & 1 & 0 & 1 \\ 2 & 1 & 2 & 1 & 2 & 1 & 2 & 2 & 1 & 0 \end{bmatrix} \quad (6)$$

$$\boldsymbol{CD3} = \begin{bmatrix} 0 & 2 & 4 & 3 & 2 \\ 1 & 1 & 3 & 3 & 3 \\ 2 & 0 & 2 & 4 & 4 \\ 3 & 1 & 1 & 3 & 3 \\ 4 & 2 & 0 & 2 & 3 \\ 3 & 3 & 1 & 1 & 2 \\ 3 & 4 & 2 & 0 & 1 \\ 2 & 4 & 3 & 1 & 0 \\ 1 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \quad (7)$$

$$\boldsymbol{f(RD3, CD3)} = \begin{bmatrix} 0.0312 & 0.0313 & 0.0024 & 0.1817 & 0.0242 \\ 0.0024 & 0.0313 & 0.0312 & 0.0242 & 0.1817 \\ 0.0182 & 0.0023 & 0.0312 & 0.0312 & 0.0312 \\ 0.0312 & 0.0023 & 0.0182 & 0.0312 & 0.0312 \\ 0.2317 & 0.2323 & 0.2317 & 0.0232 & 0.0232 \end{bmatrix} \quad (8)$$

$$\boldsymbol{RD2} = \begin{bmatrix} 3 & 3 & 1 & 0 & 1 & 2 & 3 & 2 & 2 & 1 \\ 3 & 2 & 3 & 2 & 1 & 0 & 1 & 2 & 2 & 1 \\ 1 & 3 & 1 & 2 & 3 & 2 & 3 & 0 & 2 & 1 \\ 1 & 1 & 3 & 2 & 3 & 2 & 2 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 & 2 & 1 & 2 & 1 & 1 & 0 \end{bmatrix} \quad (9)$$

$$\boldsymbol{CD2} = \begin{bmatrix} 0 & 2 & 2 & 4 & 3 \\ 2 & 0 & 4 & 3 & 1 \\ 2 & 4 & 0 & 2 & 4 \\ 3 & 3 & 1 & 1 & 3 \\ 4 & 3 & 2 & 0 & 2 \\ 3 & 2 & 3 & 1 & 1 \\ 3 & 1 & 4 & 2 & 0 \\ 1 & 3 & 1 & 3 & 3 \\ 1 & 1 & 3 & 3 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix} \quad (10)$$

$$\boldsymbol{f(RD2, CD2)} = \begin{bmatrix} 0.0024 & 0.1817 & 0.0313 & 0.0312 & 0.0242 \\ 0.0182 & 0.0312 & 0.0023 & 0.0312 & 0.0312 \\ 0.0312 & 0.0242 & 0.0313 & 0.0024 & 0.1817 \\ 0.0312 & 0.0312 & 0.0023 & 0.0182 & 0.0312 \\ 0.2317 & 0.0232 & 0.2323 & 0.2317 & 0.0232 \end{bmatrix} \quad (11)$$

As a result, the two configurations in Figs. 2 and 3 are isomorphic because they have the same sorted identity matrix SMI.

$$\boldsymbol{SM_I = sort(RD3 \times CD3) = sort(RD2 \times CD2) =}$$
$$\begin{bmatrix} 0.0023 & 0.0182 & 0.0312 & 0.0312 & 0.0312 \\ 0.0023 & 0.0182 & 0.0312 & 0.0312 & 0.0312 \\ 0.0024 & 0.0232 & 0.0312 & 0.0313 & 0.1817 \\ 0.0024 & 0.0242 & 0.0312 & 0.0313 & 0.1817 \\ 0.0232 & 0.0242 & 0.2317 & 0.2317 & 0.2323 \end{bmatrix} \quad (12)$$

Non-isomorphism identification

There is no similarity between the KCs in Fig. 3 and Fig. 4, according to the proposed method.

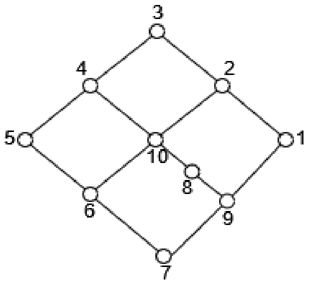


Figure 4. Ten bars KC not isomorphic to those of Fig. 2 and Fig 3

SDM2, SDM3, and SDM4 are the sorted distance matrices of the configurations shown in Figs. 2, 3, and 4.

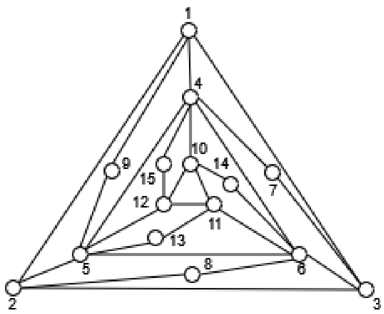
$$\boldsymbol{SDM_2 = SDM_3} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 \end{bmatrix} \quad (13)$$

$$\boldsymbol{SDM_4} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 4 \end{bmatrix} \quad (14)$$

As a consequence, the graphs in Figs. 3 and 4 have different sorted distance matrices. As a result, Figs. 3 and 4 are not isomorphic.

B. Application to a Fifteen Vertices KC

The two kinematic chains have the same sorted distance matrix, but there is no similarity between Fig. 5 and Fig 6, as confirmed by the proposed method.



(B)

Figure 5. Fifteen bars KC

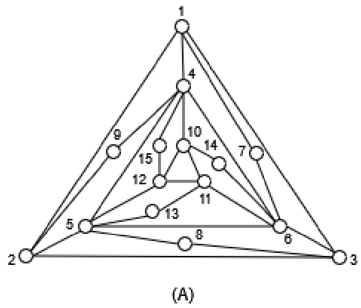


Figure 6. Fifteen bars KC non-isomorphic to that of Fig. 5

The sorted distance matrix of the kinematic chain shown in Figs. 5 and 6 is given by:

$$\boldsymbol{SDM_5 = SDM_6} = \begin{bmatrix} 0&1&1&1&1&1&1&2&2&2&2&2&2&2&2 \\ 0&1&1&1&1&1&1&2&2&2&2&2&2&2&2 \\ 0&1&1&1&1&1&1&2&2&2&2&2&2&2&2 \\ 0&1&1&1&1&2&2&2&2&2&2&3&3&3&3 \\ 0&1&1&1&1&2&2&2&2&2&2&3&3&3&3 \\ 0&1&1&1&1&2&2&2&2&2&2&3&3&3&3 \\ 0&1&1&1&1&2&2&2&2&2&2&3&3&3&3 \\ 0&1&1&1&1&2&2&2&2&2&2&3&3&3&3 \\ 0&1&1&1&1&2&2&2&2&2&2&3&3&3&3 \\ 0&1&1&2&2&2&2&2&2&3&3&3&3&3&3 \\ 0&1&1&2&2&2&2&2&2&3&3&3&3&3&3 \\ 0&1&1&2&2&2&2&2&2&3&3&3&3&3&3 \\ 0&1&1&2&2&2&2&2&2&3&3&3&3&3&3 \\ 0&1&1&2&2&2&2&2&2&3&3&3&3&3&3 \\ 0&1&1&2&2&2&2&2&2&3&3&3&3&3&3 \end{bmatrix} \quad (15)$$

The M_Is for configurations A and B, as shown in Figs. 5 and 6, are as follows:

$$\boldsymbol{M_{IA}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0017 & 0.0017 & 0.0172 & 0.0172 \\ 0.0000 & 0.0000 & 0.0017 & 0.0017 & 0.0172 & 0.0172 \\ 0.0000 & 0.0000 & 0.0017 & 0.0017 & 0.0172 & 0.0172 \\ 0.0024 & 0.0024 & 0.0024 & 0.0024 & 0.0242 & 2.4242 \\ 0.0024 & 0.0024 & 0.0024 & 0.0024 & 0.0242 & 2.4242 \\ 0.0024 & 0.0024 & 0.0024 & 0.0024 & 0.0242 & 2.4242 \\ 0.0024 & 0.0024 & 0.0024 & 0.0024 & 0.0242 & 2.4242 \\ 0.0024 & 0.0024 & 0.0024 & 0.0024 & 0.0242 & 2.4242 \\ 0.0024 & 0.0024 & 0.0024 & 0.0024 & 0.0242 & 2.4242 \end{bmatrix} \quad (16)$$

$$\boldsymbol{M_{IB}} = \begin{bmatrix} 0.0000 & 0.0000 & 0.0017 & 0.0017 & 0.0172 & 0.0172 \\ 0.0000 & 0.0000 & 0.0017 & 0.0017 & 0.0172 & 0.0172 \\ 0.0000 & 0.0000 & 0.0017 & 0.0017 & 0.0172 & 0.0172 \\ 0.0002 & 0.0024 & 0.0024 & 0.0242 & 0.0242 & 2.4242 \\ 0.0002 & 0.0024 & 0.0024 & 0.0242 & 0.0242 & 2.4242 \\ 0.0002 & 0.0024 & 0.0024 & 0.0242 & 0.0242 & 2.4242 \\ 0.0002 & 0.0024 & 0.0024 & 0.0242 & 0.0242 & 2.4242 \\ 0.0002 & 0.0024 & 0.0024 & 0.0242 & 0.0242 & 2.4242 \\ 0.0002 & 0.0024 & 0.0024 & 0.0242 & 0.0242 & 2.4242 \end{bmatrix} \quad (17)$$

C. Identification of 28 bars Configuration

Configurations A and B in Fig. 7 are isomorphic. A and C, on the other hand, are not. Vertex 28 in graph C connects to vertex 16 rather than vertex 11 in graph A.

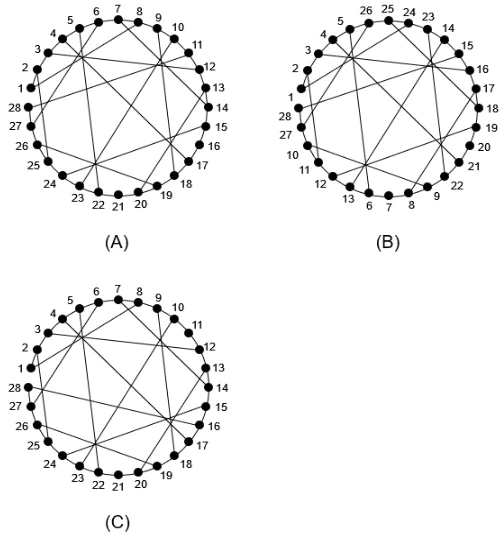


Figure 7. Three twenty-eight bars KCs

The sorted distance matrices of the graphs in Fig. 7 are as follows:

$$\boldsymbol{SDM_{A\&B}} = \begin{bmatrix} 0000000000000000000000000000 \\ 2\times 1111111111111111111111111111 \\ 1111111111111111111111112222 \\ 3\times 2222222222222222222222222222 \\ 2\times 2222222222222222222222223333 \\ 2222222222222222333333333333 \\ 5\times 3333333333333333333333333333 \\ 4\times 3333333333333333333333334444 \\ 2\times 3333333333333333444444444444 \\ 2\times 4444444444444444444444444444 \\ 4\times 4444444444444444444444445555 \\ 4444444444444444555555556666 \end{bmatrix} \quad (18)$$

$$\boldsymbol{SDM_C} = \begin{bmatrix} 0000000000000000000000000000 \\ 2\times 1111111111111111111111111111 \\ 1111111111111111111111112222 \\ 3\times 2222222222222222222222222222 \\ 2\times 2222222222222222222222223333 \\ 2222222222222222333333333333 \\ 5\times 3333333333333333333333333333 \\ 4\times 3333333333333333333333334444 \\ 3333333333333333444444444444 \\ 3333333333333444444444444444 \\ 3334444444444444444444444444 \\ 4444444444444444444444444444 \\ 2\times 4444444444444444444444445555 \\ 4444444444444444444444555555 \\ 4444444444444444444455555556 \\ 4444444444444455555555566666 \end{bmatrix} \quad (19)$$

Note: $n \times (K)$ indicates that the K column in the sorted distance matrix is duplicated n times.

Graph C is not isomorphic to graphs A and B, according to the SDMC.

The following are the identity matrices for graphs A and B:

$$\boldsymbol{M_{IA}} = \begin{bmatrix} 8\times(0.071) & 16\times(0.049) & 16\times(5.94) & 16\times(4040.4) \\ 16\times(0.049) & 8\times(4949.3) & 8\times(6059.4) & 8\times(6059.4) \end{bmatrix} \quad (20)$$

$$M_{IB} = \begin{bmatrix} 2\times(0.1) & 2\times(0.5) & 4\times(0.5) & 4\times(0.05) \\ 3\times(0.1) & 4\times(0.5) & 2\times(0.6) & 2\times(3938.3) \\ 3\times(0.1) & 2\times(0.6) & 2\times(7.1) & 2\times(4040.4) \\ 3\times(0.5) & 4\times(0.6) & 4\times(4040.4) & 4\times(4840.4) \\ 3\times(0.5) & 4\times(5.94) & 4\times(4940.4) & 4\times(4940.4) \\ 2\times(0.6) & 4\times(6) & 4\times(4949.3) & 2\times(5940.4) \\ 2\times(0.6) & 4040.3 & 2\times(5940.4) & 4\times(6049.5) \\ 3\times(5.9) & 2\times 4840.4 & 6159.4 & 6159.4 \\ 3\times(5.9) & 5050.4 & 7150.5 & 7150.5 \end{bmatrix} \quad (21)$$

Note: $n\times(K)$ indicates that K appears n times in the identity matrix columns.

Therefore, graphs A and B are isomorphic.

IV. Comparative Analysis

A. Isomorphic Identification for Kinematic Chains Using Variable High-order Adjacency Link Values

The basic idea behind this method [4] is to construct the adjacency matrix of one of two configurations under consideration by employing the correspondence manner of the pair of corresponding strings, which is calculated with the high-order adjacency link string of each vertex of the two configurations. Two isomorphic KCs' adjacency matrices are identical; otherwise, they are not. A number of steps must be taken to determine isomorphism. To begin, use the following equation to calculate each vertex's high-order adjacency link value:

$$s_i^r = s_i^{r-1} + \frac{1}{10}\sum_{j=1}^{n} s_j^{r-1} d_{i,j} \quad (22)$$

Where n denotes the number of vertices in the configuration, di,j is an element from the ith row and jth column of the adjacency matrix, and r is the maximum length of the shortest path between each specified vertex. The initial value of the vertex i is $s_i^0 = s_i$. Second, if a duplicate value is found in the string $S^r = \{s_1^r, s_2^r, \ldots, s_n^r\}$, the KC reassignment is performed [4] until a new string $S_{a,q,l}^r$ is obtained with no duplication. Third, search for string pairs that are similar in both configurations. If there are no such pairs, the two configurations are not isomorphic. Determine the correspondence form of two corresponding strings in the fourth stage. The adjacency matrix of one of the two configurations should then be computed and compared using the correspondence form to the second matrix. If the two KCs' adjacency matrices are the same, they are isomorphic; otherwise, they are not.

Application on fifteen bars configuration of Figs 5 and 6

In this case r = 3,

According to Figs 5 and 6, the 0-order adjacency link values of the two KCs A and B are:

$$S_A^0 = S_B^0 = \{2,2,2,3,3,3,1,1,1,2,2,2,1,1,1\} \quad (23)$$

According to Figs 5 and 6, the three-order adjacency link values of KCs A and B are as follows:

$$S_A^3(1,1:8) = S_B^3(1,1:8) = \{5.526, 5.526, 5.526, 8.306, 8.306, 8.306, 3.183, 3.183\} \quad (24)$$

$$S_A^3(1,9:15) = S_B^3(1,9:15) = \{3.183, 5.526, 5.526, 5.526, 3.183, 3.183, 3.183\} \quad (25)$$

Because the three-order adjacency link values of A and B have duplicate values, we must reassign KCA and KCB. After two reassignment processes, KCA becomes the following. (Time execution of this step is O(n^3)):

$$S_{A,2}^3(1,1:8) = \{8.083, 6.297, 6.269, 10.114, 8.781, 8.837, 3.858, 3.282\} \quad (26)$$

$$S_{A,2}^3(1,9:15) = \{3.614, 5.899, 5.599, 5.627, 3.220, 3.252, 3.552\} \quad (27)$$

We obtain six forms of the 3-order adjacency link value of KCB after two reassignment processes. (Time execution of this step is O(mn^2)). m is the number of edges.

One form used below as an example

$$S_{B,2}^3(1,1:8) = \{8.083, 6.269, 6.269, 10.114, 8.837, 8.781, 3.614, 3.282\} \quad (28)$$

$$S_{B,2}^3(1,9:15) = \{3.858, 5.899, 5.597, 5.629, 3.222, 3.250, 3.552\} \quad (29)$$

Because the $S_{A,2}^3$ and the six 3-order adjacency link values of KCB differ, the two configurations are not isomorphic

Fortunately, no duplicates were discovered along the six S_{B2}^3 forms. Otherwise, we must continue reassigning S_{B2}^3 until no duplicate between S_{B2}^3 forms are found In each case, the adjacency matrix of configuration B is created and compared to the adjacency matrix of configuration A. In terms of computer execution time, the proposed method (O(n^3)) in this study is unquestionably simpler and faster than the published method by Ding et al. [4] (O(n^3+mn^2)). It instantly returns the aforementioned identity matrices.

B. Re-labeling Technic

Application of the method to the KC of Fig. 6.

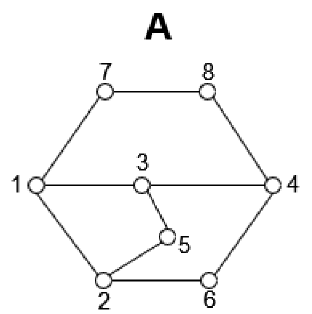


Figure 8 Eight bars KC

Isomorphism Identification by Binary Code [9]

By converting a sequence of numbers generated by the upper triangular matrix to an integer number, the adjacent matrix was used to generate the binary code from the top triangle of the matrix. This maximum integer number is unique to this configuration, but finding it requires labeling the graph configuration n! times, which yields a n! possible Adjacent matrix and a n! possible Integer number. Only the maximum (or minimum) number can be used to identify isomorphism.

Some methods have been developed to overcome the disadvantages of having so many labeling possibilities. Here we will examine two approaches.

The first method [10] searches for the perimeter loop, which is the largest cycle in the configuration; in Fig. 9, we purposefully chose a configuration with only one perimeter loop [1,2,6,4,8,7,1] to simplify the explanation. Otherwise, before comparing perimeter loops, we must perform these operations for each possible perimeter loop [10].

Starting from the highest vertex degree in the perimeter loop and proceeding clockwise and counterclockwise, we calculate the sequence of vertices degrees as 322323 and 332322, respectively. As a result, the maximum loop is the one that runs anticlockwise.

The second step is to rebuild the graph configuration by drawing the perimeter loop on the board and then relabeling the border vertices from 1 to 6 counter clockwise, as shown in Fig. 9, and then relabeling the inner loop, as shown in [10].

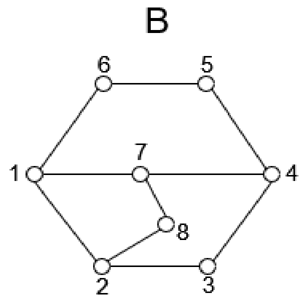


Figure 9 Fig 9's KC after relabeling

The adjacent matrix of the KC of Fig. 9 transforms to:

$$\begin{bmatrix} 0 & \{1\} & 0 & 0 & 0 & \{1\} & (1) & 0 \\ 1 & 0 & \{1\} & 0 & 0 & 0 & 0 & (1) \\ 0 & 1 & 0 & \{1\} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \{1\} & 0 & (1) & 0 \\ 0 & 0 & 0 & 1 & 0 & \{1\} & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & (1) \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (30)$$

The upper triangle matrix in the preceding matrix produces the characteristic code identification 1000110100001100001010100001. We can write the code as τ - β, where τ is the number of vertices in the perimeter loop in braces and β is the vertex's sequence position in the new adjacent matrix in parenthesis (). The first (1) in the new adjacent matrix, for example, is in the first row and seventh column, so the code is 17. The second (1) code is 28, and so on.

As a result, 617284778 is the code identification for Fig. 9.

This method takes three steps to complete: (n vertices, m edges, b perimeter loops)

- Define all possible cycles in the configuration take O(m^2)
- Find the perimeter loops O(n^3)
- - Convert the adjacent matrix O (bn^2)

The total execution time is O(m^2+n^3+bn^2)

The second method [11] relabel the graph configuration as follows:

- The starting vertex must be the vertex with the highest degree; if several vertices have the same degree, the starting vertex is the vertex adjacent to the highest degree vertices. This condition is met by vertices 1 and 3 in Fig. 8, so vertex 1 is chosen as the starting vertex.
- The remaining vertices are labeled based on their proximity to the previously named ones. For example, vertex 3 will be labeled 2 because it is adjacent to vertex 1 and has the highest degree of the vertices adjacent to vertex 1. Vertex 2 will be labeled as 3, and vertex 7 will be labeled as 4.
- Repeat step 2 until all of the vertices are labeled. Continue with vertex number 2, and so on.

This method has a time execution O(Sn^2), such as $S = \sum_{k=1}^{n} d_k^2$ d_k is the degree of vertex k and n is the number of vertices in the configuration.

The relabeled KC and the adjacent matrix are shown below in Fig. 10:

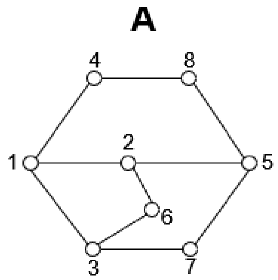


Figure 10 KC of Fig. 8 relabeled with the second method

The adjacent matrix becomes

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (31)$$

The binary max code is derived from the adjacent matrix's upper triangular part.

$(1110000001100001100001011000)_2 =$
$1\times2^{28}+1\times2^{27}+1\times2^{26}+1\times2^{19}+1\times2^{18}+1\times2^{13}+1\times2^{12}+1\times2^{7}+1\times2^{5}+1\times2^{4} = 470560944$

Both methods are ineffective when dealing with a large number of vertices. This is because identifying the perimeter loop and relabeling the vertices takes more CPU time than obtaining the adjacent matrix and generating the code identification. When there is more than one perimeter loop and many vertices in the KC have the same degree, the problem becomes more severe. Despite this, both methods are practical and simple for small numbers of vertices, but they clearly fall short of the proposed method (execution time O(n^3)), particularly for large numbers of vertices.

C. Numbering Method for the Kinematic Chain Isomorphism Recognition of Planar KCs

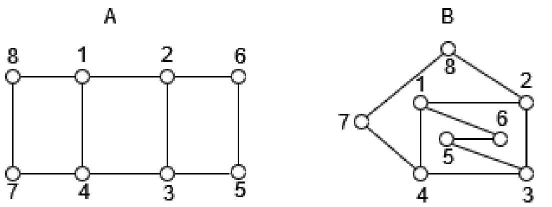


Figure 11. Eight bars kinematic chain

Qing Tian et al. [21] proposed a simple method for counting the vertices. This method employs specific rules based on the degree and weight of vertices, and it will be compared to the preceding configuration.

The procedure starts with determining the following properties:

di: represents the vertex degree. The following values are assigned to di:

- di =-1 if vertex i is binary and is connected to two non-binary vertices,
- di=-2 if i is a binary vertex linked to a binary and a non-binary vertex (e.g., d5=d6=d7=d8=-2 in Fig 11),
- When i is a non-binary vertex, di corresponds to the proper vertex degree in (Fig 11, for example, d1=d2=d3=d4=3).

I1Di: is the I-level relation code formed by concatenating, in descending order, the vertex degree (di) of adjacent vertices to vertex I.

Example 1: I1D1,2,3,4=33-2, I1D5,6,7,8=33

I2Di: is the II-level relation code formed by arranging in descending order the I1Di of the adjacent vertices to vertex i.

Example 2: I2D1,2,3,4=3333-233-2, I2D5,6,7,8=333-233-2

Each of these steps needs $O(Sn^2)$ as execution time such as $\boldsymbol{S} = \sum_{k=1}^{n} \boldsymbol{d_k}^2$ $\boldsymbol{d_k}$ is the degree of vertex k.

Si: is the relation code sum. It is the sum of the algebraic relation code of vertex i in each level (I1Di, I2Di).

Example 3: S1,2,3,4=3+3+(-2) =4, S5,6,7,8=3+3=6

This method labels the vertices configuration by comparing of di, ImDi, and Si for each vertex in the two configurations (see [23]). This comparison takes $O(n^2)$.

The total execution time is $O(Sn^2) + O(n^2) = O(Sn^2)$. However, in Fig **1**, the di, ImDi, and Si are equal for all types of vertices in both configurations (see examples 1, 2, and 3 above), making labeling the vertices in both configurations impossible. As a result, every configuration must be labeled more than n/2! times. In the example of Fig. 11, 24 different labeling for each configuration implies 24^2 comparisons between the matrices of each configuration.

By comparing the SD for both configurations, the proposed method in this paper easily detects the fact that they are not isomorphic.

$$SDM_A = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 \\ 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 0 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \end{bmatrix}$$

$$SDM_B = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 0 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \end{bmatrix}$$

Fig. 11 shows two configurations that are not isomorphic. The superiority and efficiency of the proposed method for identifying isomorphisms in KCs is demonstrated by this simple example with a small number of bars.

D. Compound Topological Invariant Based Method for Detecting Isomorphism in Planar Kinematic Chains

The CTI is calculated using the fourth power of the adjacency matrix A4 (sorting each row element in decreasing order, then comparing the rows and arranging them in decreasing order) and the eigenvalues of the configuration sorted in decreasing order [23]. Despite its simplicity, this method cannot detect isomorphism between some basic-link graphs, as shown in Fig 12, because their eigenvalues and eigenvectors differ based on the results obtained in MATLAB. The eigenvalues and eigenvectors show the differences.

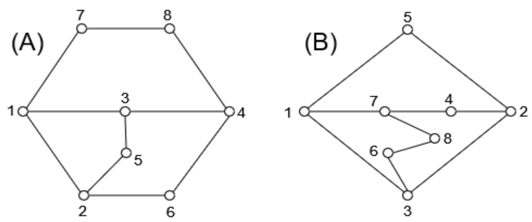


Figure 12. Two isomorphic eight bars KCs

Each column of eigenvalue and eigenvector matrices is sorted in ascending order to facilitate comparison, as shown below. The eigenvalues of A and B are as follows:

$$e_A = \begin{bmatrix} -2.281333395043705 \\ -1.944310733422072 \\ -1.140548094310296 \\ -0.099890015366104 \\ 0.598633976166785 \\ 1.000000000000000 \\ 1.275163171393258 \\ 2.592285090582134 \end{bmatrix}, e_B = \begin{bmatrix} -2.281333395043705 \\ -1.944310733422072 \\ -1.140548094310296 \\ -0.099890015366104 \\ 0.598633976166784 \\ 0.999999999999998 \\ 1.275163171393258 \\ 2.592285090582136 \end{bmatrix} \quad (34)$$

The sorted eigenvector of A and B

$$V_A = \begin{bmatrix} -0.644 & -0.500 & -0.408 & -0.175 & -0.079 & -0.016 & 0.245 & 0.323 \\ -0.526 & -0.447 & -0.400 & -0.084 & -0.047 & 0.000 & 0.273 & 0.464 \\ -0.500 & -0.412 & -0.388 & -0.080 & 0.000 & 0.164 & 0.403 & 0.488 \\ -0.473 & -0.410 & -0.252 & -0.041 & 0.106 & 0.255 & 0.405 & 0.500 \\ -0.466 & -0.409 & -0.178 & 0.000 & 0.197 & 0.266 & 0.423 & 0.579 \\ -0.453 & -0.211 & -0.110 & 0.097 & 0.240 & 0.302 & 0.440 & 0.598 \\ -0.371 & -0.098 & 0.000 & 0.183 & 0.258 & 0.379 & 0.446 & 0.630 \\ -0.113 & -0.082 & 0.156 & 0.238 & 0.360 & 0.434 & 0.500 & 0.631 \end{bmatrix} \quad (35)$$

$$V_B = \begin{bmatrix} -0.630 & -0.500 & -0.434 & -0.371 & -0.240 & -0.098 & -0.097 & 0.000 \\ -0.579 & -0.464 & -0.423 & -0.258 & -0.041 & 0.079 & 0.238 & 0.403 \\ -0.500 & -0.446 & -0.405 & -0.255 & -0.016 & 0.106 & 0.252 & 0.466 \\ -0.488 & -0.412 & -0.302 & -0.197 & 0.000 & 0.175 & 0.388 & 0.500 \\ -0.447 & -0.409 & -0.273 & -0.084 & 0.080 & 0.178 & 0.410 & 0.526 \\ -0.440 & -0.323 & -0.266 & 0.000 & 0.110 & 0.211 & 0.453 & 0.598 \\ -0.400 & -0.245 & -0.156 & 0.047 & 0.113 & 0.360 & 0.473 & 0.631 \\ -0.379 & -0.183 & 0.000 & 0.082 & 0.164 & 0.408 & 0.500 & 0.644 \end{bmatrix} \tag{36}$$

As a result, the eigenvalues and eigenvectors of the two graphs differ. MI below shows how the proposed method recognizes the similarity between graphs A and B in Fig. 12. (The sorted distance matrices of the two kinematic chains are identical).

$$MI_{A\&B} = \begin{bmatrix} 0.0012 & 0.0171 & 0.0171 & 0.0172 \\ 0.0017 & 0.0171 & 0.0171 & 0.0172 \\ 0.0121 & 0.0171 & 0.0172 & 0.1712 \\ 0.0171 & 0.0182 & 0.0242 & 1.3121 \end{bmatrix} \tag{37}$$

In terms of efficiency, as demonstrated by the preceding example, the CTI method cannot be compared to the proposed method.

V. Conclusion

Isomorphism in kinematic chains is the most critical problem in mechanism design and synthesis. This problem was solved using a simple and efficient distance matrix-based method. As shown in the comparative section, this method outperforms other methods published in recent years in detecting isomorphism for a variety of examples. Furthermore, in the previous section, efficiency was discussed, and the proposed method was shown to work for complex kinematic chains.

This method will be improved in future work to identify the isomorphism of kinematics chains with different types of joints between each two connected bars in the configuration.

Nomenclature

CDX: Columns of binary vertices
CPVS: Cubic descending PVS
DM: Distance matrix
G: Matrix obtained from the product of incident matric
KCSM: Kinematic Chain Structural Matrix
KCs: Kinematic Chains
MI: Identity Matrix
PVS: Power Vertex Similarity
RDX: Rows of non-binary vertices
SDM: Sorting distance matrix
SGf: Final Sequence of matrix G
SMI: Sorting identity matrix

Conflict Interests

The authors declare that they have not competing interests

Authors' Contribution

Mr Mohamed Aly Abdel Kader carried out the research, conceived of the presented idea, conducted the research, analyzed the data and wrote the manuscript. Prof Abdeslam Aannaque supervised the project, provided feedback and guidance, and proofread the manuscript; all authors had approved the final version.

References

[1] V. Dharanipragada and M. Chintada, "Split hamming string as an isomorphism test for one degree-of-freedom planar simple-jointed kinematic chains containing sliders," J. Mech. Des, vol. 138, no. 8, p. 082301, August 2016.
[2] V. V. Kamesh, K. M. Rao, and A. B. S. Rao, "An innovative approach to detect isomorphism in planar and geared kinematic chains using graph theory," J. Mech. Des., vol. 139, no. 12, p. 122301, December 2017
[3] W. J. Yang and H. Ding, "The perimeter loop based method for the automatic isomorphism detection in planetary gear trains," J. Mech. Des, vol. 140, no. 12, p. 123302, December 2018
[4] L. Y. He, F. X. Lin, L. Sun, and C. Y. Wu, "Isomorphic identification for kinematic chains using variable high-order adjacency link values," J. Mech. Sc. Tech, vol. 33, no. 10, pp. 1-9, 2019.
[5] A. Dwivedi, A. Bhattacharjee, and J. N. Yadav, "A method to detect isomorphism in planar kinematic chains," Machine. Mechanism and Robotics: Conference Series 978-981-16-0550-5_48
[6] W. Sun, J. Kong, and L. Sun, "A joint–joint matrix representation of planar kinematic chains with multiple joints and isomorphism identification," Adv. Mech. Engineering, vol. 10, no. 6, pp. 1–10, 2018.
[7] Z. Y. Chang, C. Zhang, Y. H. Yang, and Y. X. Wang, "A new method to mechanism kinematic chain isomorphism identification," Mech. Mach. Theory, vol. 37, pp. 411–417, 2002
[8] F. Yang, Z. Deng, J. Tao, and L. Li, "A new method for isomorphism identification in topological graphs using incident matrices," Mech. Mach. Theory, vol. 49, pp. 298–307, 2012.
[9] M. Helal, J. W. Hu, and H. Eleashy, "A new algorithm for unique representation and isomorphism detection of planar kinematic chains with simple and multiple joints," Processes, vol. 9, p. 601, 2021
[10] H. F. Ding, P. Huang, W. J. Yang, and A. Kecskeméthy, "Automatic generation of the complete set of planar kinematic chains with up to six independent loops and up to 19 links," Mech. Mach. Theory, vol. 96, pp. 75–93, 2016
[11] R. K. Rai and S. Punjabi, "A new algorithm of links labelling for the isomorphism detection of various kinematic chains using binary code," Mech. Mach. Theory, vol. 131, pp. 1–32, 2019
[12] L. Yu, H. Wang, and S. Zhou, "Graph isomorphism identification based on link-assortment adjacency matrix," *Sādhanā*, vol. 47, p. 151, 2022
[13] R. J. Cui, Z. Ye, S. Xu, C. Wu, and L. Sun, "Synthesis of planar kinematic chains with prismatic pairs based on a similarity recognition algorithm," J. Mechanisms Robotics, October 2021, vol. 13, no. 5, p. 051001.
[14] L. Sun, Z. Ye, R. Cui, X. Huang, and C. Wu, "Eliminating isomorphism identification method for synthesizing nonfractionated kinematic chains based on graph similarity," Mech. Mach. Theory, vol. 167, p. 104500, 2022.
[15] W. Sun, J. Kong, and L. Sun, "The improved hamming number method to detect isomorphism for kinematic chain with multiple joints," J. Adv. Mech. Des. Syst. Manuf, vol. 11, 2017.
[16] H. Eleashy, "A new atlas for 8-bar kinematic chains with up to 3 prismatic pairs using joint sorting code," Mech. Mach. Theory, vol. 124, pp. 118–132, 2018
[17] W. J. Yang, H. F. Ding, and A. Kecskeméthy, "Automatic synthesis of plane kinematic chains with prismatic pairs and up to 14 links," Mech. Mach. Theory, vol. 132, pp. 236–247, 2019
[18] T. Deng, H Xu, P. Tang, P. Liu, and L. Yan, "A novel algorithm for the isomorphism detection of various kinematic chains using topological index," Mech. Mach. Theory, vol. 146, p. 103740, 2020.
[19] L. Sun, R J. Cui, Z. Z. Ye, Y. Z. Zhou, Y. D. Xu, and C. Y. Wu, "Similarity recognition and isomorphism identification of planar kinematic chains," Mech. Mach. Theory, vol. 145, p. 103678, 2020.
[20] R. K. Rai and S. Punjabi, "Kinematic chains isomorphism identification using link connectivity number and entropy neglecting tolerance and clearance," Mech. Mach. Theory, vol. 123, pp. 40–65, 2018.

[21] Q. Tian, X. H. Wei, C. Li, G. Liu, and Y. M. Deng "Numbering method for the kinematic chain isomorphism recognition of a planar mechanism," Journal of Physics Conference Series, 1983 (2021) 012028

[22] S. H. Rizvi, A. Hasan, and R. A. Khan, "An efficient algorithm for distinct inversions and isomorphism detection in kinematic chains," Perspectives in Science, vol. 8, pp. 251-253, 2016.

[23] P. R. He, W.J. Zhang, Q. Li, and F. X. Wu, "A new method for detection of graph isomorphism based on the quadratic form," ASME J. Mech. Des, vol. 125, pp. 640–642, 2003

[24] H. F. Ding and Z. Huang, "A unique representation of the kinematic chain and the atlas database," Mech. Mach. Theory, vol. 42, pp. 637–651, 2007.

[25] H. F. Ding and Z. Huang, "A new theory for the topological structure analysis of kinematic chains and its applications," Mech. Mach. Theory, vol. 42, pp. 1264–1279, 2007.

[26] H. F. Ding and Z. Huang, "The establishment of the canonical perimeter topological graph of kinematic chains and isomorphism identification," ASME J. Mech. Des, vol. 129, pp. 915–923, 2007.

[27] H. F. Ding and Z. Huang, "Isomorphism identification of graphs: Especially for the graphs of kinematic chains," Mech. Mach. Theory, vol. 44, pp. 122–139, 2009.

[28] L. Sun, Z. Ye, R. Cui, W. Yang, and C. Wu, "Compound topological invariant based method for detecting isomorphism in planar kinematic chains," ASME J. Mech. Des, vol. 12, p. 054504-1, 2020.