Original Article

Institution of MECHANICAL ENGINEERS

Journal of MECHANICAL ENGINEERING SCIENCE

Proc IMechE Part C:
J Mechanical Engineering Science
2021, Vol. 235(20) 4988–4997

DOI: 10.1177/0954406220980507
journals.sagepub.com/home/pic
SAGE

Algorithm for computing the connectivity in planar kinematic chains and its application

Wenjian Yang[1], Huafeng Ding[2] and Andres Kecskemethy[1]

Abstract
The number of synthesized kinematic chains usually is too large to evaluate individual characteristics of each chain. The concept of connectivity is useful to classify the kinematic chains. In this paper, an algorithm is developed to automatically compute the connectivity matrix in planar kinematic chains. The main work is to compute two intermediate parameters, namely the minimum mobility matrix and the minimum distance matrix. The algorithm is capable of dealing with both simple-jointed and multiple-jointed kinematic chains. The present work can be used to automatically determine kinematic chains satisfying the required connectivity constraint, and is helpful for the creative design of mechanisms. The practical application is illustrated by taking the face-shovel hydraulic excavator for instance.

Keywords
Connectivity, connectivity matrix, planar kinematic chain, creative design, mechanism

Date received: 24 March 2020; accepted: 24 October 2020

Introduction

Structure synthesis of kinematic chains is an effective way to derive novel mechanisms with excellent performances.[1,2] During the last several decades, numerous methods have been developed to synthesize various kinds of kinematic chains, including simple-jointed[3–9] and multiple-jointed chains.[10–15] The number of synthesized kinematic chains is usually so large that it is hard or impossible to evaluate individual characteristics of each chain. Therefore, in practical application, some parameters are used to classify the kinematic chains, and restrict the number of chains required to be considered. This work is useful for selecting suitable kinematic chains and enhancing the efficiency in the design of mechanisms.

In the initial stage of designing mechanisms, the most discussed parameters for classifying kinematic chains include the number of links, the mobility of the kinematic chain, the number of joints suitable to be selected as actuated joints, the type of kinematic joints, the degree of the frame link and the degree of the end-effector, etc. This paper will discuss another important parameter called the connectivity. In a given kinematic chain, the connectivity $C_{i,\ j}$ between links i and j is defined as the relative mobility between links i and j. For some kinds of mechanisms such as the face-shovel hydraulic excavator,[16] the main motion mechanism of forging manipulator[17,18] and the parallel robot,[19,20] the connectivity between the frame link and the end-effector should be equal to a specified value. Therefore, the connectivity is useful for selecting the candidate kinematic chains satisfying the required constraint. The connectivity determines the ability of the end-effector to perform a task relative to the frame link.

The connectivity between two links of an open kinematic chain can be easily measured as the mobility of all the associated joints between the two links.[20] For this reason, the studies for computing the connectivity in literature mainly focused on closed kinematic chains.

Tischler et al.[19,21] discussed the relationship between the connectivity and variety, and enumerated the kinematic chains that are matched to a predefined task on the basis of variety. Shoham and Roth[20] developed the adding virtual edge method to calculate the connectivity in planar and spatial mechanisms,

[1]Institute of Mechatronics and System Dynamics, University of Duisburg-Essen, Duisburg, Germany
[2]School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, China

Corresponding author:
Huafeng Ding, School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China.
Email: dhf@ysu.edu.cn

and attempted to derive the manipulators whose frame link and end-effector have the connectivity of six. Belfiore and Di Benedetto[22] developed an automatic procedure to compute the connectivity in spatial kinematic chains. Liberati and Belfiore[23] developed a procedure to identify whether a kinematic chain has partial mobility, and developed an algorithm called "circuits gradual freezing" to compute the connectivity in planar and spatial kinematic chains. Martins and Carboni[24] redefined the concepts of connectivity in an algorithmic form, and computed the connectivity in planar kinematic chains. Huang et al.[25] developed a mathematical formula to compute the connectivity in planar kinematic chains.

The studies for computing the connectivity in literature only focused on simple-jointed kinematic chains. It should be mentioned that, apart from simple-jointed kinematic chains, multiple-jointed kinematic chains which possess the merits of minimizing space requirement and weight, reducing the number of polygonal links and simplifying the kinematic analysis, have also been used in various mechanical systems.[11–15,18] However, the analysis of connectivity in multiple-jointed kinematic chains to the knowledge of the authors has not been discussed in literature. The main purpose of this paper is to develop an automatic algorithm for computing the connectivity in planar closed kinematic chains, including both simple-jointed and multiple-jointed kinematic chains.

Basic concepts

Simple-jointed and multiple-jointed kinematic chains

If all the joints of a kinematic chain are binary, the kinematic chain is a simple-jointed kinematic chain. For example, Figure 1(a) shows the mechanism of the face-shovel hydraulic excavator in Ding et al.,[16] and Figure 1(b) shows the corresponding kinematic chain which is a 12-link 3-DOF (degree-of-freedom) simple-jointed kinematic chain. A multiple-jointed kinematic chain is a chain containing at least one multiple joint. For example, Figure 2(a) shows the main motion mechanism of the forging manipulator in Ref. [18], and Figure 2(b) shows the corresponding kinematic chain which is a 14-link 3-DOF multiple-jointed kinematic chain. Joints J1 and J2 in Figure 2(b) are multiple (ternary) joints. A *d*-nary multiple joint is equivalent to (*d*-1) revolute joints.

Single-color and bicolor topological graphs

A simple-jointed kinematic chain can be represented by a single-color topological graph where a solid vertex denotes a link and an edge denotes a joint. A multiple-jointed kinematic chain can be represented by a bicolor topological graph where the additional

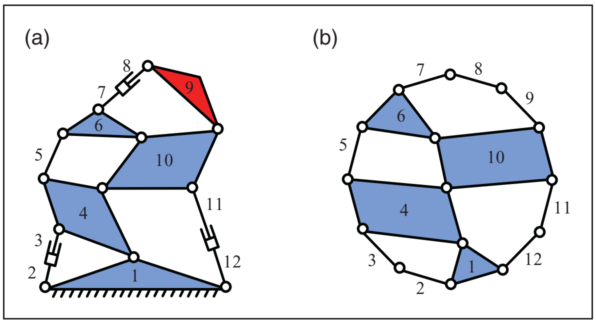


Figure 1. (a) The mechanism of a face-shovel hydraulic excavator and (b) its kinematic chain.

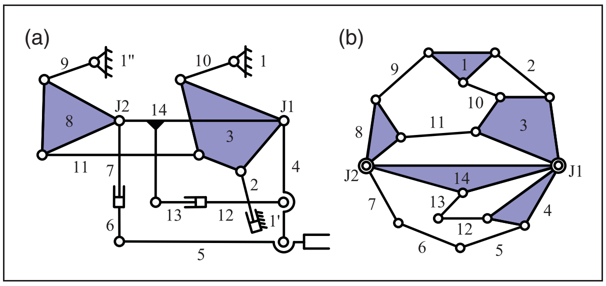


Figure 2. (a) The main motion mechanism of a forging manipulator and (b) its kinematic chain.

multiple joint is denoted by a hollow vertex.[14] For example, the single-color topological graph of Figure 1(b) is shown in Figure 3(a), and the bicolor topological graph of Figure 2(b) is shown in Figure 3 (b), where hollow vertices 15 and 16 denote multiple joints J1 and J2, respectively.

Fractionated and non-fractionated graphs

If a graph can be separated into two independent sub-graphs at a vertex, this vertex is called a cut vertex and this graph is said to be vertex-fractionated; if a graph can be separated into two independent sub-graphs at an edge, this edge is called a bridge and this graph is said to be edge-fractionated.[26] For example, Figure 4(a) shows a vertex-fractionated graph where vertex 3 is a cut vertex, and Figure 4(b) shows an edge-fractionated graph where edge 5-11 is a bridge. If a graph cannot be separated into two independent sub-graphs at any vertex or edge, this graph is said to be non-fractionated. For example, the graphs shown in Figure 3 are non-fractionated.

The algorithm for computing the connectivity

In a given kinematic chain, each link has a connectivity relative to each other link. The connectivity between each pair of links can be denoted by the connectivity matrix. In this section, an algorithm is

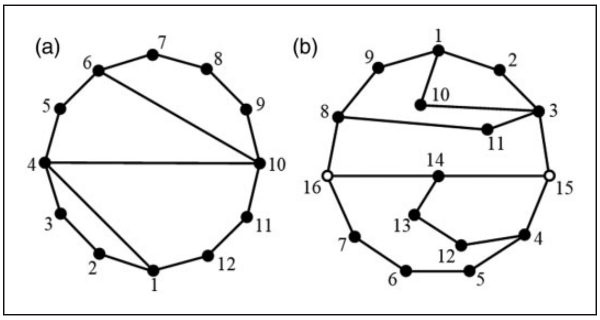


Figure 3. The topological graphs corresponding to Figures 1 (b) and 2(b).

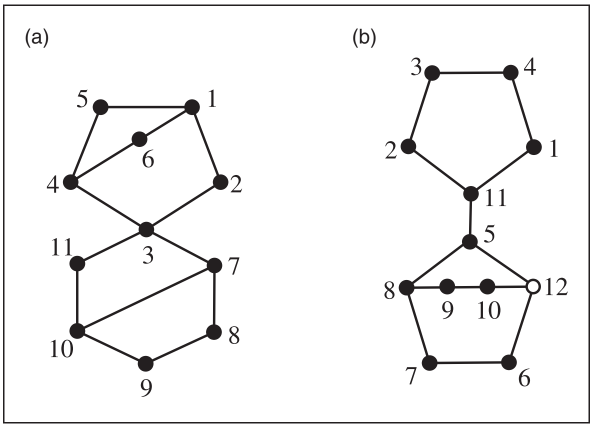


Figure 4. Fractionated graphs.

developed to compute the connectivity matrix in planar kinematic chains. The algorithm has been automated with the aid of computer programming language C++. Figure 3(a) is used to interpret the algorithm step by step.

Step 1 Compute the minimum mobility matrix.

Step 1.1 Determine the minimum independent loops in the given topological graph.

According to Euler's equation, the number of independent loops is $r = E - N + 1$, where E is the number of edges and N is the total number of vertices (including solid and hollow vertices) in the topological graph. An array of minimum independent loops is denoted as $\{L_1, L_2 \ldots L_r\}$. Minimum independent loops are determined according to the following two rules.

Rule 1: Each minimum independent loop contains as few solid vertices as possible.

Rule 2: Each minimum independent loop cannot be obtained by combining other minimum independent loops.

In Figure 3(a), the number of independent loops is $r = 4$. The minimum independent loops are determined as $\{L_1 = 1\text{-}2\text{-}3\text{-}4,\ L_2 = 1\text{-}4\text{-}10\text{-}11\text{-}12,\ L_3 = 4\text{-}5\text{-}6\text{-}10,\ L_4 = 6\text{-}7\text{-}8\text{-}9\text{-}10\}$.

Step 1.2 Acquire non-fractionated sub-chains by combining the minimum independent loops, and compute the mobility (namely DOF) of each sub-chain. For the four minimum independent loops in Figure 3(a), there are 2^4 linear combinations resulting in (2^4–1) sub-chains (there is one null combination). Then, fractionated sub-chains and repetitive sub-chains are eliminated. Ten non-fractionated sub-chains in Figure 3(a) can be acquired, as shown in Figure 5. According to Grübler equation, the mobility of a kinematic chain is defined by $F = 3(n_s - 1) - 2R$, where n_s and R are the numbers of links (solid vertices) and revolute joints in a sub-chain, respectively. Taking Figure 5(e) for instance, we have $n_s = 7$ and $R = 8$; hence, its mobility is $F = 3 \times (7-1) - 2 \times 8 = 2$. The mobility of each sub-chain is shown in Figure 5.

Step 1.3 Determine all the sub-chains containing two specified solid vertices i and j. The minimum mobility of these sub-chains is signed as $M_{i,j}$. If there exists no any sub-chain containing solid vertices i and j, let $M_{i,\ j}$ be equal to the mobility of the given topological graph (this situation occurs only when the given topological graph is fractionated). For vertices 1 and 5 in Figure 3(a), the sub-chains containing the two vertices are Figure 5(f), (h), (i) and (j). The minimum mobility of the four sub-chains is 2, thus we have $M_{1,5} = 2$.

Step 1.4 Acquire the minimum mobility matrix $M = (M_{i,j})_{n \times n}$, where $M_{i,\ j}$ is the element in i-th row and j-th column of the matrix, and n is the number of solid vertices in the given topological graph. All matrices involved in this paper are symmetric and the diagonal elements are equal to zero. The minimum mobility matrix $(M_{i,j})_{12 \times 12}$ of Figure 3(a) is automatically computed and shown in the upper-right sub-window of Figure 6.

Step 2 Compute the minimum distance matrix.

Step 2.1 Acquire the weighted adjacency matrix $W = (W_{i,j})_{n \times n}$. The weighted distance $W_{i,\ j}$ between solid vertices i and j is determined according to the following equation.

$$W_{i,\ j} = \begin{cases} 0, \text{ if i is equal to j;} \\ 1, \text{ if vertex i is adjacent to vertex j;} \\ 1, \text{ if vertices i and j are adjacent to} \\ \quad \text{the same hollow vertex;} \\ \infty, \text{ otherwise} \end{cases} \tag{1}$$

In equation (1), the third case only occurs in the multiple-jointed topological graph. For example, in Figure 3(b), solid vertices 3 and 4 are adjacent to hollow vertex 15. We should add a virtual edge between vertices 3 and 4 and set $W_{3,4} = W_{4,3} = 1$. The notation "∞" means an infinite value. In the computer program, it can be set as a considerably large number such as 50. The weighted adjacency matrix $(W_{i,j})_{12 \times 12}$ of Figure 3(a) is shown in Figure 7.

Step 2.2 For each two solid vertices i and j in a sub-chain whose mobility is equal to 1, if $W_{i,j} \neq 1$, add a virtual edge between vertices i and j, then set $W_{i,j} = 1$

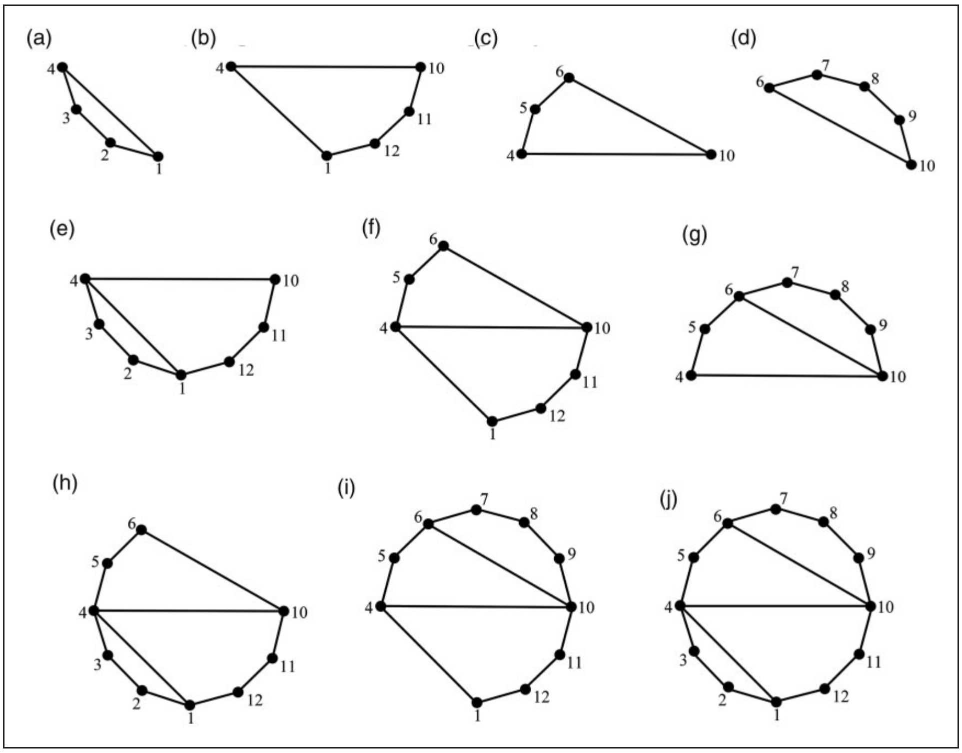


Figure 5. Non-fractionated sub-chains in Figure 3(a).

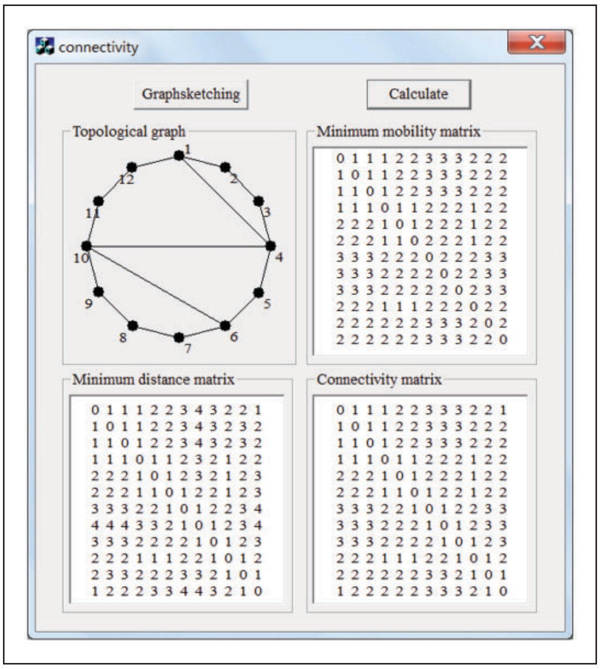


Figure 6. Automatic computation of connectivity matrix in Figure 3(a).

and acquire the modified weighted adjacency matrix W'.

The sub-chains in Figure 5(a) and (c) have the mobility of 1. For Figure 5(a), vertices 1 and 3 are not adjacent, as well as vertices 2 and 4, thus $W_{1,3} \neq 1$ and $W_{2,4} \neq 1$. As show in Figure 8(a), a virtual edge is added between vertices 1 and 3, as well as vertices 2 and 4. For the modified weighted adjacency matrix W' in Figure 8(b), elements $W_{1,3}$ and $W_{2,4}$ are set as "1", and the symmetric elements $W_{3,1}$ and $W_{4,2}$ are also set as "1". Similarly, by considering the sub-chain in Figure 4(c), elements $W_{4,6}$, $W_{6,4}$, $W_{5,10}$ and $W_{10,5}$ in Figure 8(b) are set as "1".

Step 2.3 Define an array of recursive matrices D^0, D^1 … D^k … D^n where $D^k = (D^k_{i,j})_{n\times n}$. Set $D^0 = W'$ and compute D^1 … D^k … D^n in turn according to the recursive formula in equation (2).

$$D^k_{i,j} = \min:\{\ D^{k-1}_{i,j},\ D^{k-1}_{i,k} + D^{k-1}_{k,j}\} \tag{2}$$

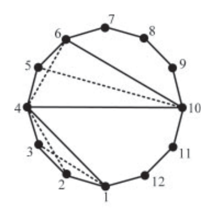
The formula in equation (2) is known as Floyd Algorithm, which is frequently utilized to compute the minimum distance in graphs. Using this formula, the n-th recursive matrix $D^n = (D^n_{i,j})_{n\times n}$ is the minimum distance matrix $D = (D_{i,\ j})_{n\times n}$ of the topological graph. Here, $D_{i,j}$ is the minimum distance between solid vertices i and j (virtual edges are included).

Continuing the example, the initial matrix D^0 is equal to the matrix W' shown in Figure 8(b). Then, compute the first recursive matrix $D^1 = (D^1_{i,j})_{12\times 12}$ according to the formula $D^1_{i,j} = \min:\{D^0_{i,j}, D^0_{i,1} + D^0_{1,j}\}$, and compute the second recursive matrix $D^2 = (D^2_{i,j})_{12\times 12}$ according to the formula $D^2_{i,j} = \min:\{D^1_{i,j}, D^1_{i,2} + D^1_{2,j}\}$, and so on. For example, we have $D^1_{1,2} = \min:\{D^0_{1,2}, D^0_{1,1} + D^0_{1,2}\} = \min:\{1, 0+1\} = 1$, and $D^1_{2,12} = \min:\{D^0_{2,12}, D^0_{2,1} + D^0_{1,12}\} = \min:\{\infty, 1+1\} = 2$. The first recursive matrix D^1 is shown

$$W=\begin{bmatrix}0&1&\infty&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1\\1&0&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty\\\infty&1&0&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty\\1&\infty&1&0&1&\infty&\infty&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&1&0&1&\infty&\infty&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty\\\infty&\infty&\infty&1&\infty&1&\infty&\infty&1&0&1&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0&1\\1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0\end{bmatrix}$$

Figure 7. The weighted adjacency matrix of Figure 3(a).

(a)



(b)

$$W'=\begin{bmatrix}0&1&1&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1\\1&0&1&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty\\1&1&0&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty\\1&1&1&0&1&1&\infty&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&1&0&1&\infty&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&1&1&0&1&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty\\\infty&\infty&\infty&1&1&1&\infty&\infty&1&0&1&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0&1\\1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0\end{bmatrix}$$

Figure 8. (a) The graph in Figure 3(a) with adding virtual edges and (b) the modified weighted adjacency matrix.

$$D^1=\begin{bmatrix}0&1&1&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1\\1&0&1&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&2\\1&1&0&1&\infty&\infty&\infty&\infty&\infty&\infty&\infty&2\\1&1&1&0&1&1&\infty&\infty&\infty&1&\infty&2\\\infty&\infty&\infty&1&0&1&\infty&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&1&1&0&1&\infty&\infty&1&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0&1&\infty&\infty\\\infty&\infty&\infty&1&1&1&\infty&\infty&1&0&1&\infty\\\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&\infty&1&0&1\\1&2&2&2&\infty&\infty&\infty&\infty&\infty&\infty&1&0\end{bmatrix}$$

Figure 9. The first recursive matrix derived from Figure 8(b).

in Figure 9. The 12-th recursive matrix is the minimum distance matrix $D=(D_{i,j})_{12\times 12}$, as shown in the lower-left sub-window of Figure 6. The element $D_{i,j}$ is the minimum distance between solid vertices i and j. For example, element $D_{2,5}$ of the minimum distance matrix in Figure 6 is equal to 2. This value corresponds to the path 2-4-5 which has two edges in Figure 8(a).

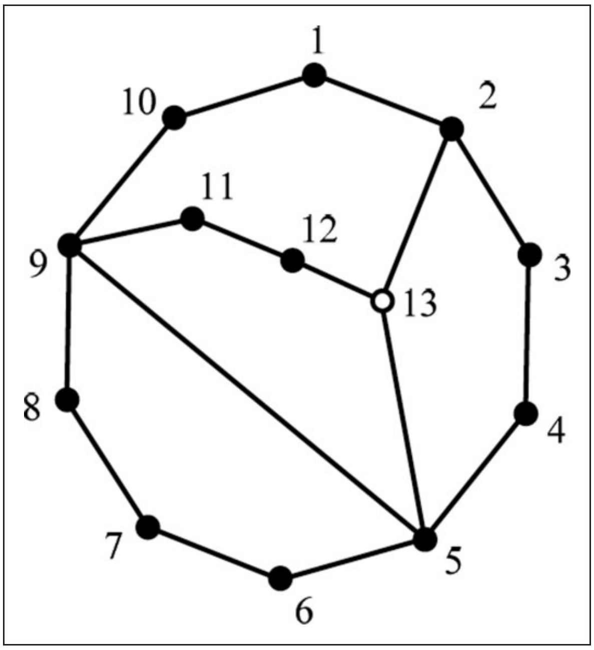


Figure 10. A 12-link 3-DOF multiple-jointed kinematic chain.

Step 3 Compute the connectivity matrix $C=(C_{i,\ j})_{n\times n}$ according to the formula in equation (3), where λ is the order of the screw system ($\lambda=3$ for planar kinematic chains).

$$C_{i,j}=\min:\left\{M_{i,j},D_{i,j},\lambda\right\} \tag{3}$$

According to the minimum mobility matrix $M=(M_{i,\ j})_{12\times 12}$ acquired in Step 1, the minimum distance matrix $D=(D_{i,\ j})_{12\times 12}$ acquired in Step 2 and $\lambda=3$, the connectivity matrix $C=(C_{i,\ j})_{12\times 12}$ of Figure 3(a) can be acquired, as shown in the lower-right sub-window of Figure 6. The connectivity between the frame link (link 1) and the end-effector (link 9) is $C_{1,9}=3$.

Computation of the connectivity in a multiple-jointed kinematic chain

The following process with considering Figure 10 is used to prove the ability of our algorithm for dealing with multiple-jointed kinematic chains.

Step 1 Compute the minimum mobility matrix. The graph in Figure 10 has four independent loops. Its minimum independent loops are determined as $\{L_1=$2-3-4-5-13, $L_2=$5-9-11-12-13, $L_3=$1-2-13-5-9-10, $L_4=$5-6-7-8-9$\}$. Fourteen non-fractionated sub-chains can be acquired and shown in Figure 11. The minimum mobility of all the sub-chains containing solid vertices i and j is signed as $M_{i,\ j}$. For example, considering vertices 2 and 5, the sub-chains containing the two vertices are Figure 11(a), (c), (e), (f), (g), (i), (j), (k), (l), (m) and (n). The minimum mobility of these sub-chains is 1, thus we have $M_{2,5}=1$. The minimum mobility matrix $(M_{i,\ j})_{12\times 12}$ of Figure 10 is

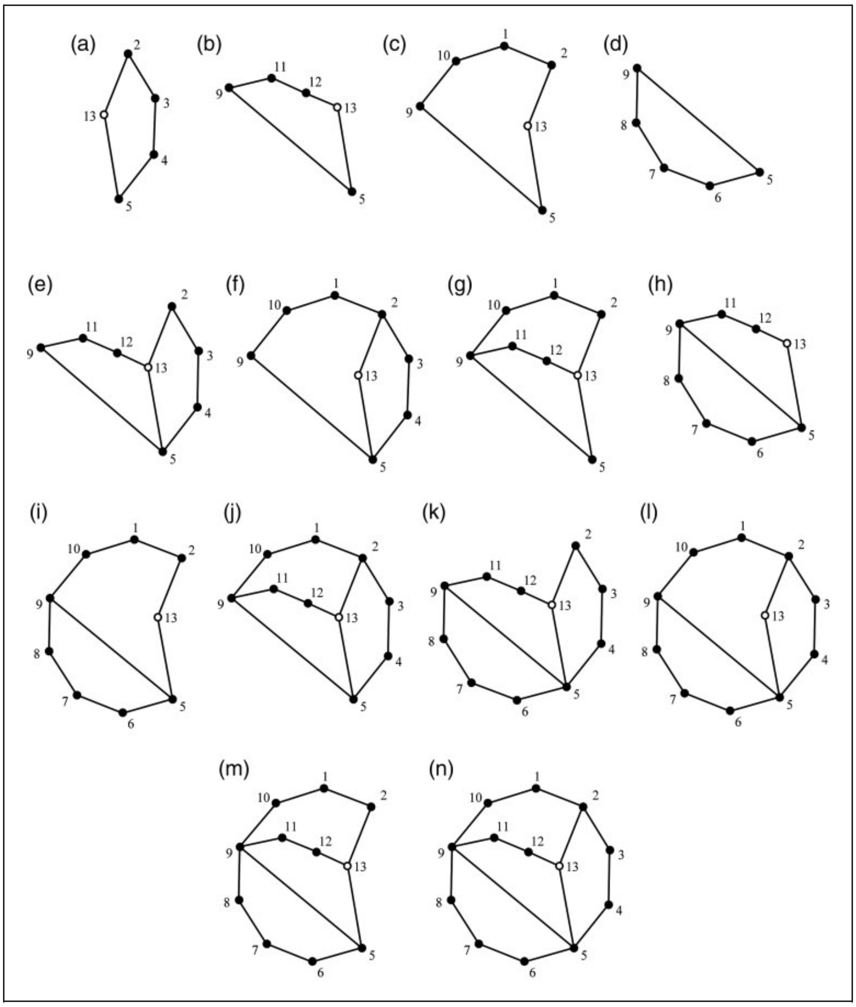


Figure 11. Non-fractionated sub-chains in Figure 10.

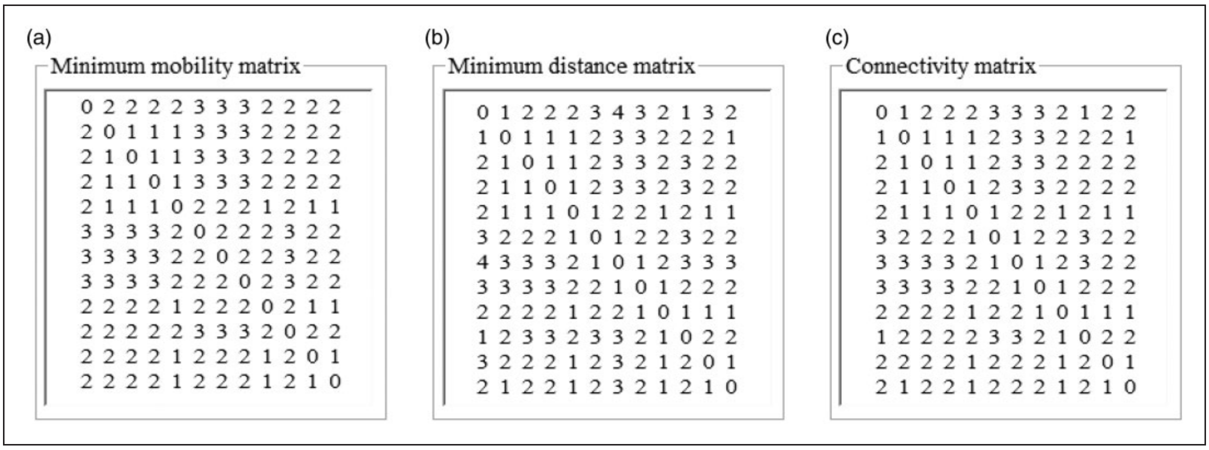


Figure 12. Automatic computation of connectivity matrix of Figure 10.

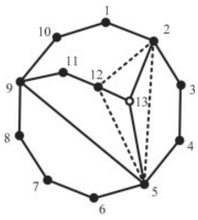
automatically computed and shown in Figure 12(a). In order to save space, the software interface is not displayed.

Step 2 Compute the minimum distance matrix.

Step 2.1 Vertices 2 and 5 are adjacent to hollow vertex 13, hence vertices 2 and 5 are connected with a virtual edge. Edges 2-13 and 5-13 are equivalent to one revolute joint, hence the distance between vertices 2 and 5 is equal to 1. Similarly, vertices 2 and 12, and 5 and 12 are, respectively, connected with a virtual edge, as shown in Figure 13(a). According to equation (1), the weighted adjacency matrix $W=(W_{i,j})_{12\times 12}$ of Figure 10 is acquired and shown in Figure 13(b). Due to the addition of virtual edges, we have $W_{2,5} = W_{5,2} = W_{2,12} = W_{12,2} = W_{5,12} = W_{12,5} = 1$.

Step 2.2 The sub-chains in Figure 11(a) and (b) have the mobility of one, hence vertices 3 and 5, 2 and 4, 5 and 11, and 9 and 12 are, respectively, connected with a virtual edge, as shown in Figure 14(a).

(a)

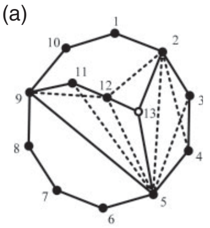


(b)

$$W = \begin{bmatrix} 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty & \infty \\ 1 & 0 & 1 & \infty & 1 & \infty & \infty & \infty & \infty & \infty & \infty & 1 \\ \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 1 & \infty & 1 & 0 & 1 & \infty & \infty & 1 & \infty & \infty & 1 \\ \infty & \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 1 & \infty & \infty & 1 & 0 & 1 & 1 & \infty \\ 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty & 0 & 1 \\ \infty & 1 & \infty & \infty & 1 & \infty & \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix}$$

Figure 13. The weighted adjacency matrix of Figure 10.

(a)



(b)

$$W' = \begin{bmatrix} 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & \infty & \infty \\ 1 & 0 & 1 & 1 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & 1 \\ \infty & 1 & 0 & 1 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 1 & 1 & 0 & 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\ \infty & 1 & 1 & 1 & 0 & 1 & \infty & \infty & 1 & \infty & 1 & 1 \\ \infty & \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 1 & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 1 & \infty & \infty & 1 & 0 & 1 & 1 & 1 \\ 1 & \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & 1 & \infty & \infty & \infty & 1 & \infty & 0 & 1 \\ \infty & 1 & \infty & \infty & 1 & \infty & \infty & \infty & 1 & \infty & 1 & 0 \end{bmatrix}$$

Figure 14. The modified weighted adjacency matrix of Figure 10.

The modified weighted adjacency matrix W' is shown in Figure 14(b).

Step 2.3 The initial matrix D^0 is equal to the matrix W' shown in Figure 14(b). Compute D^1 ...... D^k ...... D^{12} in turn according to the recursive formula in equation (2). The 12-th recursive matrix is the minimum distance matrix $D = (D_{i,j})_{12 \times 12}$ of Figure 10, as shown in Figure 12(b). The element $D_{i,j}$ is the minimum distance between solid vertices i and j (virtual edges are included). For example, the element $D_{1,5}$ of the minimum distance matrix in Figure 12(b) is equal to 2. This value corresponds to the path 1-2-5 in Figure 14(a).

Step 3 According to the formula in equation (3), the connectivity matrix $C = (C_{i,j})_{12 \times 12}$ of Figure 10 is shown in the Figure 12(c).

To verify the validity, the present algorithm has been used to the kinematic chains discussed in literature.[23–25] The results of the connectivity matrix are completely consistent with the existing results. Our results for multiple-jointed kinematic chains are new. The main work of our algorithm is to compute two intermediate parameters, namely the minimum mobility matrix and the minimum distance matrix. Minimum independent loops are combined to derive sub-chains and minimum mobility matrix. In order to derive the minimum distance matrix, solid vertices adjacent to the same hollow vertex are connected with a virtual edge, and solid vertices in 1-DOF sub-chains are also connected with a virtual edge. Compared to the previous method,[25] the present method is simpler, and possesses the ability to deal with both simple-jointed and multiple-jointed kinematic chains.

Table 1. Structural characteristics of the face-shovel hydraulic excavator.

No.	Structural characteristics	No.	Structural characteristics
1	The mechanism is planar and closed	4	The mechanism has 10 or 12 links
2	The mechanism is non-fractionated	5	The mechanism has three DOFs
3	The mechanism is simple-jointed or multiple-jointed	6	The mechanism only has revolute and prismatic joints

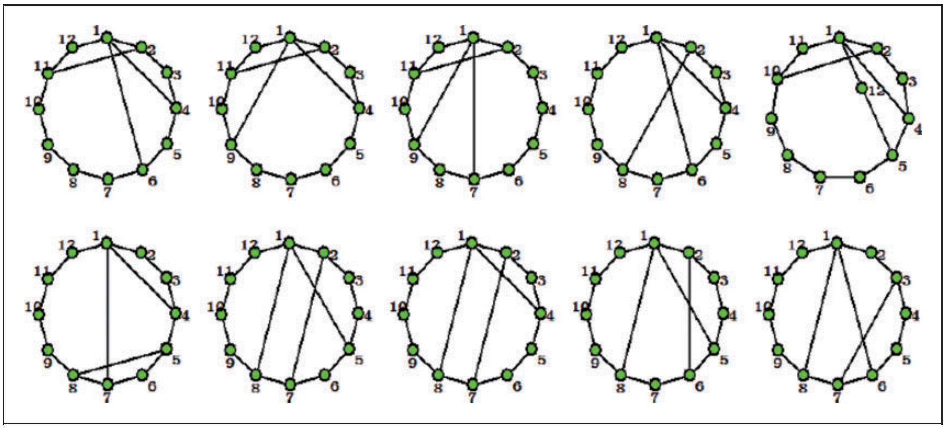


Figure 15. A part of the 1962 planar 12-link 3-DOF kinematic chains.

An example of the practical application

In our previous work, some methods have been developed to automatically synthesize simple-jointed kinematic chains[9] and multiple-jointed kinematic chains.[13,14] The present algorithm can be used to derive suitable mechanisms based on the synthesis results. This section aims to interpret how the present work is used in the design of mechanisms. The main steps of the conceptual design of a specified mechanism are illustrated as followed.

Step 1 Determine the structural characteristics of the specified mechanism, and acquire all possible kinematic chains based on the synthesis results.

Let us take the face-shovel hydraulic excavator for instance (one valid mechanism is shown in Figure 1). The structural characteristics of the excavator are listed in Table 1. Here, we only consider 12-link simple-jointed mechanisms. According to Ding et al.,[9] there are 1962 planar simple-jointed non-fractionated 12-link 3-DOF kinematic chains. A part of the 1962 topological graphs sketched by the software in Ding et al.[9] are shown in Figure 15.

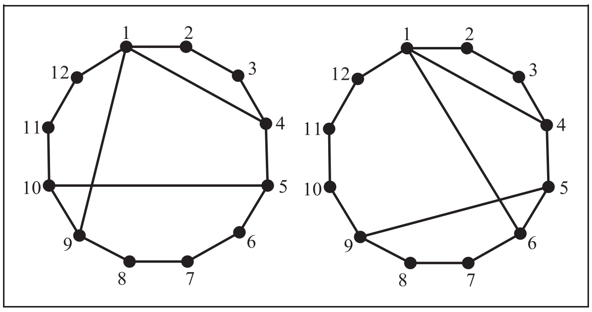


Figure 16. Two 12-link 3-DOF kinematic chains satisfying the fundamental constraints.

Step 2 Determine the kinematic chains satisfying the fundamental constraint.

A binary path is defined as a path whose starting and ending vertices are non-binary, and the other middle vertices are binary. A binary path having k middle vertices is called a k-length binary path. For example, the path 6-7-8-9-10-11 in the first graph in Figure 15 is a 4-length binary path.

The face-shovel hydraulic excavator is actuated by three separate hydraulic cylinders. The end-effector which is a binary link is directly actuated by one of the three hydraulic cylinders. Therefore, the topological graph suitable for application as the excavator should contain at least two 2-length binary paths and one 3-length binary path. The kinematic chains satisfying this fundamental constraint can be easily selected from the 1,96,212-link 3-DOF kinematic chains. For example, two kinematic chains satisfying this constraint are depicted in Figure 16.

Step 3 For each kinematic chain, determine the frame link, end-effector and actuated joints.

The frame link and end-effector of the face-shovel hydraulic excavator should satisfy the following constraints: (1) The frame link is a non-binary link, (2) the inversions having different frame links are non-isomorphic, (3) the frame link and the end-effector are not in the same minimum independent loop, and

Table 2. The selection of the frame link and end-effector for the first graph in Figure 16.

	The selection of frame link		The selection of end-effector	
No.	Constraint (1)	Constraint (2)	Constraint (3)	Constraint (4)
Case 1	Link 1	Link 1	Links 6 and 8	Link 6
Case 2	Link 4	Link 4	Links 6 and 8	Link 8
Case 3	Link 5	Link 5	None	None
Case 4	Link 9	Link 9	None	None
Case 5	Link 10	Link 10	None	None

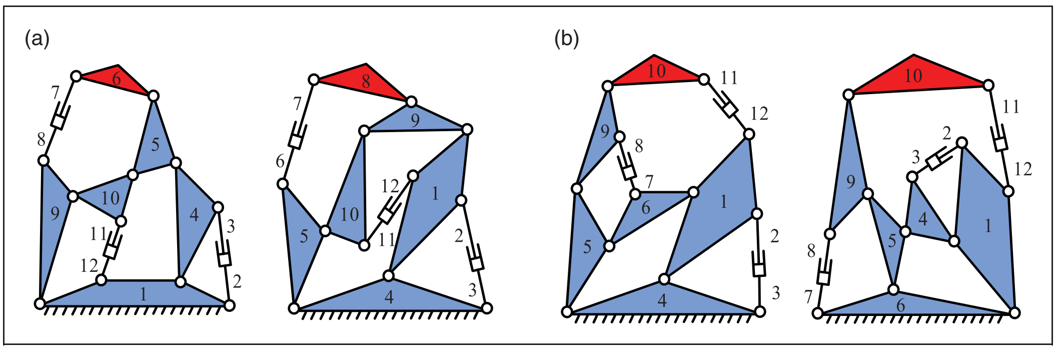


Figure 17. The valid mechanisms derived from Figure 16.

(4) the frame link and the end-effector have the connectivity of three.

Take the first graph in Figure 16 for instance. Links 1, 4, 5, 9 and 10 are non-binary, and the inversions by selecting the five links as frame links are non-isomorphic. Therefore, links 1, 4, 5, 9 and 10 can be selected as the frame link, as shown in Table 2. The end-effector is a binary link in the 3-length binary path. Only links 6 and 8 can be the end-effector. According to constraint (3), the cases 3, 4 and 5 in Table 2 are invalid, because links 5, 9, 10, 6 and 8 are in the same minimum independent loop 5-6-7-8-9-10. Let us consider case 1 in Table 2. Using the present algorithm, the connectivity between links 1 and 6 is $C_{1,6} = 3$, and the connectivity between links 1 and 8 is $C_{1,8} = 2$. According to constraint (4), only link 6 can be selected as the end-effector. Similarly, in case 2, only link 8 can be selected as the end-effector. Therefore, two valid mechanisms can be derived from the first graph in Figure 16, as shown in Figure 17(a). In the first mechanism, link 1 is the frame link and link 6 is the end-effector. In the second mechanism, link 4 is the frame link and link 8 is the end-effector. Similarly, two valid mechanisms can be derived from the second graph in Figure 16, as shown in Figure 17(b).

The present algorithm can be used to determine mechanisms satisfying the required connectivity constraint. The mechanisms having excellent performances can be derived by conducting the dimension computation, kinematic analysis and dynamic analysis.

Conclusions

The parameter of connectivity can serve as a constraint to derive mechanisms that are matched to a specified task. In this paper, an automatic algorithm is developed to compute the connectivity matrix in planar closed kinematic chains. The main work is to compute two intermediate parameters, namely the minimum mobility matrix and the minimum distance matrix. Minimum independent loops are combined to derive sub-chains and minimum mobility matrix, and Floyd Algorithm is used to derive minimum distance matrix. Solid vertices adjacent to the same hollow vertex are connected with a virtual edge, and solid vertices in 1-DOF sub-chains are also connected with a virtual edge.

The algorithm is applicable for both simple-jointed and multiple-jointed kinematic chains. Whereas, the existing methods only focused on simple-jointed kinematic chains. The practical application of the algorithm is illustrated by taking the face-shovel hydraulic excavator for instance. The algorithm can be used to derive both simple-jointed and multiple-jointed mechanisms that are matched to a specified task.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors are grateful to the projects supported by the Natural Science Foundation of China (No. 51675495) and Central Innovation Program for the Medium Size Industry (No. ZF4047825RP9).

References

1. Mruthyunjaya TS. Kinematic structure of mechanisms revisited. *Mech Mach Theory* 2003; 38: 279–320.
2. Yan HS and Chiu YT. On the number synthesis of kinematic chains. *Mech Mach Theory* 2015; 89: 128–144.
3. Davies T and Crossley F. Structural analysis of plane linkages by Franke's condensed notation. *J Mech* 1966; 1: 171–184.
4. Tuttle E. Generation of planar kinematic chains. *Mech Mach Theory* 1996; 31: 729–748.
5. Butcher EA and Hartman C. Efficient enumeration and hierarchical classification of planar simple-jointed kinematic chains: application to 12- and 14-bar single degree-of-freedom chains. *Mech Mach Theory* 2005; 40: 1030–1050.
6. Sunkari RP and Schmidt LC. Structural synthesis of planar kinematic chains by adapting a Mckay-type algorithm. *Mech Mach Theory* 2006; 41: 1021–1030.
7. Simoni R, Carboni AP and Martins D. Enumeration of kinematic chains and mechanisms. *Proc IMechE, Part C: J Mechanical Engineering Science* 2009; 223: 1017–1024.
8. Yan HS and Chiu YT. An improved algorithm for the construction of generalized kinematic chains. *Mech Mach Theory* 2014; 78: 229–247.
9. Ding HF, Huang P, Yang WJ, et al. Automatic generation of the complete set of planar kinematic chains with up to six independent loops and up to 19 links. *Mech Mach Theory* 2016; 96: 75–93.
10. Mruthyunjaya TS. Structural synthesis by transformation of binary chains. *Mech Mach Theory* 1979; 14: 221–231.
11. Yan HS and Hsu CH. Contracted graphs of kinematic chains with multiple joints. *Math Comput Model* 1988; 10: 681–695.
12. Hsu CH. Enumeration of basic kinematic chains with simple and multiple joints. *J Franklin Inst* 1992; 329: 775–789.
13. Ding HF, Yang WJ, Huang P, et al. Automatic structural synthesis of planar multiple joint kinematic chains. *ASME J Mech Des* 2013; 135: 091007–091001.
14. Ding HF, Yang WJ, Zi B, et al. The family of planar kinematic chains with two multiple joints. *Mech Mach Theory* 2016; 99: 103–112.
15. Pozhbelko V and Ermoshina E. Number structural synthesis and enumeration process of all possible sets of multiple joints for 1-dof up to 5-loop 12-link

mechanisms on base of new mobility equation. *Mech Mach Theory* 2015; 90: 108–127.
16. Ding HF, Han L, Yang WJ, et al. Kinematics and dynamics analyses of a new type face-shovel hydraulic excavator. *Proc IMechE, Part C: J Mechanical Engineering Science* 2017; 231: 909–924.
17. Chu XB, Gao F and Ge H. Direct kinematic analysis of a heavy-payload forging manipulator in the grasping stage. *Industrial Robot* 2012; 39: 304–311.
18. Ding HF, Feng ZM, Fu HY, et al. Conceptual design of an exactly straight lifting forging manipulator. *ASME J Mech Robot* 2014; 6: 034502–034501. 4.
19. Tischler CR, Samuel AE and Hunt KH. Kinematic chains for robot hands: part 2 kinematic constraints, classification, connectivity and actuation. *Mech Mach Theory* 1995; 30: 1217–1239.
20. Shoham M and Roth B. Connectivity in open and closed loop robotic mechanisms. *Mech Mach Theory* 1997; 32: 279–294.
21. Tischler CR, Samuel AE and Hunt KH. Selecting multi-freedom multi-loop kinematic chains to suit a given task. *Mech Mach Theory* 2001; 36: 925–938.
22. Belfiore NP and Di Benedetto A. Connectivity and redundancy in spatial robots. *Int J Robot Res* 2000; 19: 1245–1261.
23. Liberati A and Belfiore NP. A method for the identification of the connectivity in multi-loop kinematic chains: analysis of chains with total and partial mobility. *Mech Mach Theory* 2006; 41: 1443–1466.
24. Martins D and Carboni AP. Variety and connectivity in kinematic chains. *Mech Mach Theory* 2008; 43: 1236–1252.
25. Huang P, Ding HF, Yang WJ, et al. An automatic method for the connectivity calculation in planar closed kinematic chains. *Mech Mach Theory* 2017; 109: 195–219.
26. Martins D, Simoni R and Carboni AP. Fractionation in planar kinematic chains: reconciling enumeration contradictions. *Mech Mach Theory* 2010; 45: 1628–1641.

Appendix

Notation

$C_{i,j}$	the connectivity between links i and j
C	the connectivity matrix
D	the minimum distance matrix
$D_{i,j}$	the minimum distance between solid vertices i and j
E	the number of edges
$M_{i,j}$	the minimum mobility of the sub-chains containing vertices i and j
M	the minimum mobility matrix
n	the number of solid vertices
r	the number of independent *loops*
$W_{i,j}$	the weighted distance between solid vertices i and j
W	the weighted distance matrix
W	the modified weighted distance matrix
λ	the order of the screw system

DOI: 10.1177/0954406220980507
URN: urn:nbn:de:hbz:465-20230825-082758-7