Han *et al. Chin. J. Mech. Eng.* (2020) 33:2
https://doi.org/10.1186/s10033-019-0424-9

Chinese Journal of Mechanical Engineering

ORIGINAL ARTICLE

Open Access

A Computational Synthesis Approach of Mechanical Conceptual Design Based on Graph Theory and Polynomial Operation

Lin Han, Geng Liu*, Xiaohui Yang and Bing Han*

Abstract

The design synthesis is the key issue in the mechanical conceptual design to generate the design candidates that meet the design requirements. This paper devotes to propose a novel and computable synthesis approach of mechanisms based on graph theory and polynomial operation. The graph framework of the synthesis approach is built firstly, and it involves: (1) the kinematic function units extracted from mechanisms; (2) the kinematic link graph that transforms the synthesis problem from mechanical domain into graph domain; (3) two graph representations, i.e., walk representation and path representation, of design candidates; (4) a weighted matrix theorem that transforms the synthesis process into polynomial operation. Then, the formulas and algorithm to the polynomial operation are presented. Based on them, the computational flowchart to the synthesis approach is summarized. A design example is used to validate and illustrate the synthesis approach in detail. The proposed synthesis approach is not only supportive to enumerate the design candidates to the conceptual design of a mechanical system exhaustively and automatically, but also helpful to make that enumeration process computable.

Keywords: Design synthesis, Conceptual design, Graph theory, Polynomial operation, Kinematic link graph, Weighted matrix theorem

1 Introduction

As an early and the most creative phase of engineering design, the basic task of conceptual design is developing and synthesizing the building blocks to generate meaningful concept solutions that meet design requirements. In all the feasible design candidates, the optimal and novel one is pursued in this design phase. However, because of the limitation of bias and shorten of knowledge or experienced to the designers, it is always difficult to identify all the feasible design candidates and furthermore to find out the optimal one in them. Therefore, numerous researchers have focused their study on the computer aided design synthesis, and many models and approaches have been developed.

Based on physical working principles, Kota proposed a Function-Structure model and implemented it in the design of hydraulic systems [1]. Besides, a matrix methodology to the design synthesis was developed by him and Chiou on the basis of that model [2, 3]. Another two famous models are Function-Behavior-Structure and Function-Behavior-State [4–6]. The common character of these functional models is that they describe objects or design problems and solutions in terms of their known functions, and regard the design process as a function decomposition process. Therefore, the associated synthesis approaches are so-called function-based synthesis [7–11]. There exist other kinds of synthesis approaches, e.g., the grammar-based synthesis [12–16] and the graph-based synthesis [17–21]. In grammar-based synthesis, the generative grammars, which are a class of production systems that capture design knowledge by defining a vocabulary and rule-set, are constructed and used to generate design alternatives [22]. In graph-based

*Correspondence: npuliug@nwpu.edu.cn; npuhanbing@nwpu.edu.cn
Shaanxi Engineering Laboratory for Transmissions and Controls, Northwestern Polytechnical University, Xi'an 710072, China

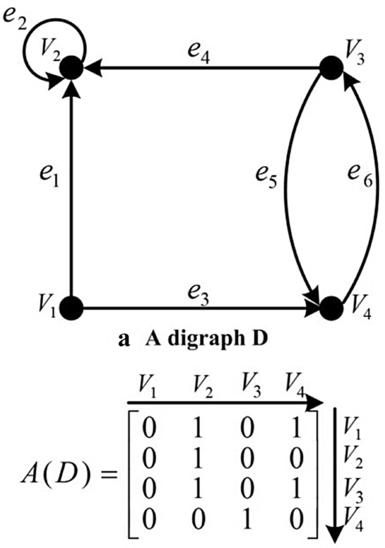
synthesis, the graph theory is used to represent a product and define the relationships between its components, and the graph concepts and theorems are employed to generate design candidates [23]. For now, graph-based synthesis approaches have been widely used in the synthesis, analysis and optimization process of linkage systems [24–26] and epicyclic gear trains [27, 28]. Moreover, Bin proposed a computational conceptual design synthesis model based on space matrix [29], and he also integrated the functional synthesis of mechanisms with mechanical efficiency and cost to find out the optimal solution [30, 31]. Similarly, Masakazu proposed an integrated optimization for supporting functional and layout designs during conceptual design phase [32].

With the help of above considerable researches, the paper proposes a novel and more computable synthesis approach of mechanisms. The advantage of this approach is that it builds up the mathematical structure of the design synthesis of mechanisms based on graph theory, and makes the design candidates are calculated by the induced polynomial operation formulas. Besides, each design candidate is finally represented by a matrix form, which will be conducive to the analysis and optimization of the design candidates in the future research. The rest of the paper is outlined as follows. Section 2 describes some basic concepts of the graph theory that will be used in the proposed synthesis approach. Section 3 builds up the graph framework of the synthesis approach. The polynomial operations including the polynomial-walk operation, edge sequence operation and vertex sequence operation are presented in Section 4. The computational flowchart of the synthesis approach is summarized in Section 5. In Section 6, the proposed synthesis approach is applied to figure out all the feasible design candidates of a mechanical system. Some concluding remarks are made in Section 7.

2 Basic Concepts on the Used Graph Theory

As a branch of mathematics, graph theory has gotten numerous applications in many fields of engineering for its ability to concisely represent and handle the relationships between different objects. In this section, some fundamental concepts of graph theory that are essential for the paper proposed synthesis approach are introduced briefly. The detailed descriptions of these concepts can be found in Ref. [33].

The mathematical structure of a directed graph or digraph D, as illustrated in Figure 1(a), is an ordered triple $(V(D), E(D), \Psi_D)$. Where $V(D)$ is a nonempty set of vertices, e.g., $\{V_1, V_2, V_3, V_4\}$, $E(D)$ is a set of edges, e.g., $\{e_1, e_2, e_3, e_4, e_5, e_6\}$, that disjoints from $V(D)$, and ψ_D is an incidence function that assigns every edge e an initial vertex $init(e)$ and a terminal vertex $ter(e)$. The edge e is said to be directed from $init(e)$ to $ter(e)$, e.g., the direction of e_1 is from $init(e_1)=V_1$ to $ter(e_1)=V_2$. Usually, an edge is always written as a two-tuple style, e.g., the two-tuple of e_3 is $e_3(V_1,V_4)$.



a A digraph D

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

b The adjacency matrix of D

Figure 1 A digraph and its adjacency matrix

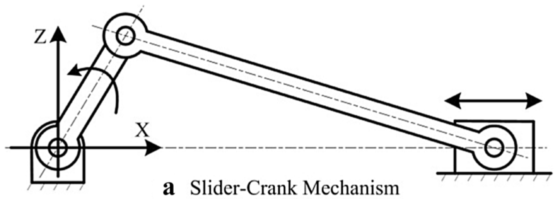
In a digraph, if the vertices of several edges are the same, such edges are called *multiple edges*, e.g., e_5 and e_6; if an edge whose initial vertex $init(e)$ equals to its terminal vertex $ter(e)$, such edge is called a *loop*, e.g., e_2. A directed walk W in D is a finite non-null sequence, e.g., $W=(V_1, e_3, V_4, e_6, V_3, e_5, V_4, e_6, V_3, e_4, V_2)$, whose terms are alternately vertices and edges. A walk is called a path if all the vertices and edges are distinct, e.g., $P=(V_1, e_3, V_4, e_6, V_3)$, and the length of it is the number of edges between the starting and ending vertices. A path of D is also a subgraph of D, for all its vertices and edges contained in D.

The topological structure, or in other words the connection relationships between vertices, can be conveniently represented in matrix form. A vertex-to-vertex adjacency matrix to a digraph is defined as $A(D) = [a_{ij}]$, where $a_{ij} = \mu(V_i, V_j)$ is the number of directed-edges whose $init(e)=V_i$ and $ter(e)=V_j$, e.g., Figure 1(b). Let $A^k(D)$ be the kth power of $A(D)$, where k is a positive integer number. Then an adjacency matrix theorem can be described as follows.

Adjacency Matrix Theorem: The number of walks of length n from vertex V_i to vertex V_j in a digraph D is given by the (i, j) element of A^n (D).

For example, the third power of $A(D)$ in Figure 1(b) is

$$A^3(D) = \begin{bmatrix} 0 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.$$



a Slider-Crank Mechanism

$$KFU_1 = \{SC_1^{RT},(R,1,-1),(T,1,1)\}$$

$$KFU_2 = \{SC_1^{TR},(T,1,1),(R,1,-1)\}$$

b Two KFUs of Slider-Crank

Figure 2 Representation of a slider-crank and its KFUs

Then, the number of walks of length 3 from V_1 to V_2 is $A^3(D)(1, 2) = 2$, and they are $W_1 = (V_1, e_1, V_2, e_2, V_2, e_2, V_2)$ and $W_2 = (V_1, e_3, V_4, e_6, V_3, e_4, V_2)$.

3 Graph Framework

3.1 Kinematic Function Unit

A mechanical system, which is made up by variety mechanisms, is designed to transform the power or motion from the system input to the system output. Therefore, each mechanism has the function of transferring and transforming the power or motion that pass through it. In this paper, the function representation of a mechanism is called a kinematic function unit (KFU). It has four fundamental features: (1) structure type (crank-rocker, spur gear, etc.), (2) motional type (rotation, translation or swing), (3) continuous or intermittent motion, (4) reciprocating motion or not. The relationship between a specific mechanism and the KFU belonging to it is one-to-one or one-to-many, for the kinematic transformation between the input and output members of a specific mechanism may be interchangeable.

For example, if the crank is the driving member in a slider-crank mechanism displayed in Figure 2(a), the slider-crank is able to transform the crank's continuous rotational motion into the slider's reciprocating translational motion. While, if slider is the driving member, the slider-crank is able to transform the slider's reciprocating translational motion into the crank's continuous rotational motion. Therefore, the slider-crank mechanism has two different KFUs. To describe these KFUs in a general way and make it easy to be applied and recognized in computer, a general mathematical structure of KFU is given by

$$KFU = \{C_i^F, MI, MO\}, \tag{1}$$

where C_i^F is the identification of a KFU, and it includes three different symbols, i.e., $\boldsymbol{C}$, $\boldsymbol{F}$, $\boldsymbol{i}$, that have diverse values and meanings. The value of $\boldsymbol{C}$ is the abbreviation for the name of a mechanism to which KFU is subordinate, e.g., *SC* in Figure 2(b) is the abbreviation of slider-crank. The value of $\boldsymbol{F}$ is the motional transformation type of a KFU, e.g., $F = RT$ in Figure 2(b) represents a rotational motion is transformed into a translational motion, while $F = TR$ represents a translational motion is transformed into a rotational motion. The value of $\boldsymbol{i}$ is a positive number and represents the serial number of a KFU in a design candidate generated through the synthesis process, in case a KFU is used more than once in that solution. The initial value of it to all KFUs are set to 1. MI and MO are two 1×3 row vectors and represent the input and output motions of the KFU respectively. The elements $x1$, $x2$ and $x3$ of these vectors are encoded symbols or numerals, and the encoded values and meanings of them are illustrated in Table 1.

3.2 Kinematic Link Graph

The kinematic link graph (KLG) is a directed graph and constructed by looking each KFU as a vertex and the kinematic relationship between two KFUs as an edge. Through KLG, the synthesis problem of mechanisms are transformed from mechanical domain into graph domain, so that the feasible design candidates can be represented and calculated by the methods and theorems of graph theory.

When building a KLG, the first task is to extract KFUs from mechanisms. Table 2 illustrates *eight* KFUs from *seven* mechanisms. Here, the input and output of a mechanism system are abstracted as two virtual mechanisms that named as "System Input" and "System Output" respectively. Specially, it should be noticed that "System Input" only has the function to send out the power or motion, while "System Output" only has the function to incept power or motion. Thus, the values of MI of "System Input" as well as the values of MO of "System Output" are set to (0, 0, 0). Since the input and output motions of a mechanism system are various according to

Table 1 Meanings and coded values to the elements *x*1, *x*2 and *x*3 of vectors MI and MO

Elements	Meanings	Coded values
*x*1	Rotational motion	R
	Translational motion	T
	Swinging motion	S
*x*2	Continuous motion	1
	Intermittent motion	−1
*x*3	Reciprocating motion	1
	Non-reciprocating motion	−1

Table 2 Eight KFUs from seven mechanisms

Mechanisms	KFUs
Crank-rocker	$\{CR_1^{RS}, (R, 1, -1), (S, 1, 1)\}$
	$\{CR_1^{SR}, (S, 1, 1), (R, 1, -1)\}$
Spur gear	$\{SG_1^{RR}, (R, 1, -1), (R, 1, -1)\}$
Geneva wheel	$\{GW_1^{RR}, (R, 1, -1), (R, -1, -1)\}$
Cam-follower	$\{CF_1^{RT}, (R, 1, -1), (T, 1, 1)\}$
Pawl-Ratchet wheel	$\{PRW_1^{SR}, (S, 1, 1), (R, -1, -1)\}$
System input	$\{SI_1^R, (0, 0, 0), (R, 1, -1)\}$
System output	$\{SO_1^R, (R, -1, -1), (0, 0, 0)\}$

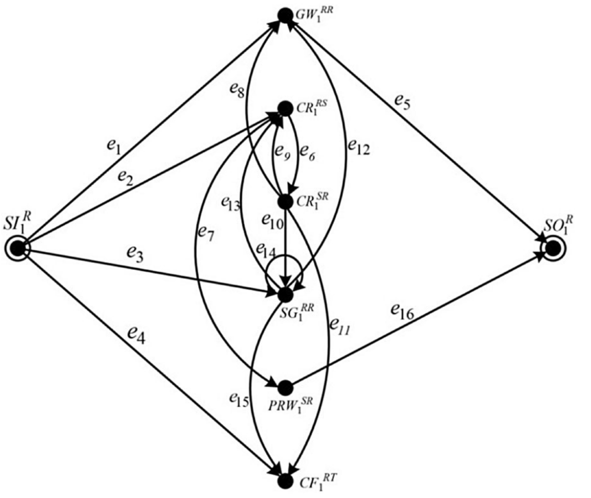


Figure 3 Graph representation of KLG(T2)

different design requirements, the KFUs of them illustrated in Table 2 are only for a simple instance.

The second task when building a KLG is to establish the rules about how the KFUs connect with each other to form the directed edges. Here, two connection rules are defined and presented below.

Connection Rule 1: If one KFU's MO is equal to another KFU's MI, a directed edge will be formed between them and the direction is from former to latter.

Connection Rule 2: If a KFU'MI is equal to its MO, a directed loop will be formed around it.

Based on the KFUs illustrated in Table 2 and the two connection rules, a kinematic-link graph is built and named as KLG(T2) as Figure 3 shows. The labels of vertices in KLG(T2) are the identifications of the corresponding KFUs. Specially, the vertices that represent the KFUs of "System Input" and "System Output" in a KLG are called *input-vertex* (e.g., SI_1^R) and *output-vertex* (e.g., SO_1^R), respectively. To distinguish them with other vertices, they are marked by "◉" in KLG(T2).

3.3 Graph Representation of Design Candidate

There are two different graph representations of design candidate, i.e., the walk representation and the path representation. They have different application backgrounds and meanings in the synthesis process.

3.3.1 Walk Representation

The walk representation adopts the vertex sequence and edge sequence of a walk to represent the mechanisms and their kinematic relationships in a design candidate. This kind of representation is mainly used in the calculation part of synthesis process.

The walk representation directly derives from KLG, for each walk whose head is input-vertex and tail is output-vertex in a KLG can be looked as a design candidate. For instance, Figure 4(a) displays a walk in KLG(T2) and the design candidate it represents is shown in Figure 4(b). Basically, the constitute of a walk can be divided into four parts: (I) vertex terms (VT), (II) edge terms (ET), (III) vertex sequence (VS), and (IV) edge sequence (ES). Where, VT and ET are two unordered sets that made up by the vertices and edges of a walk respectively, while VS and ES are the ordered sets of VT and ET respectively. Thus, the mathematical model of walk representation is defined as

$$W = \{VS(W), ES(W)\}. \tag{2}$$

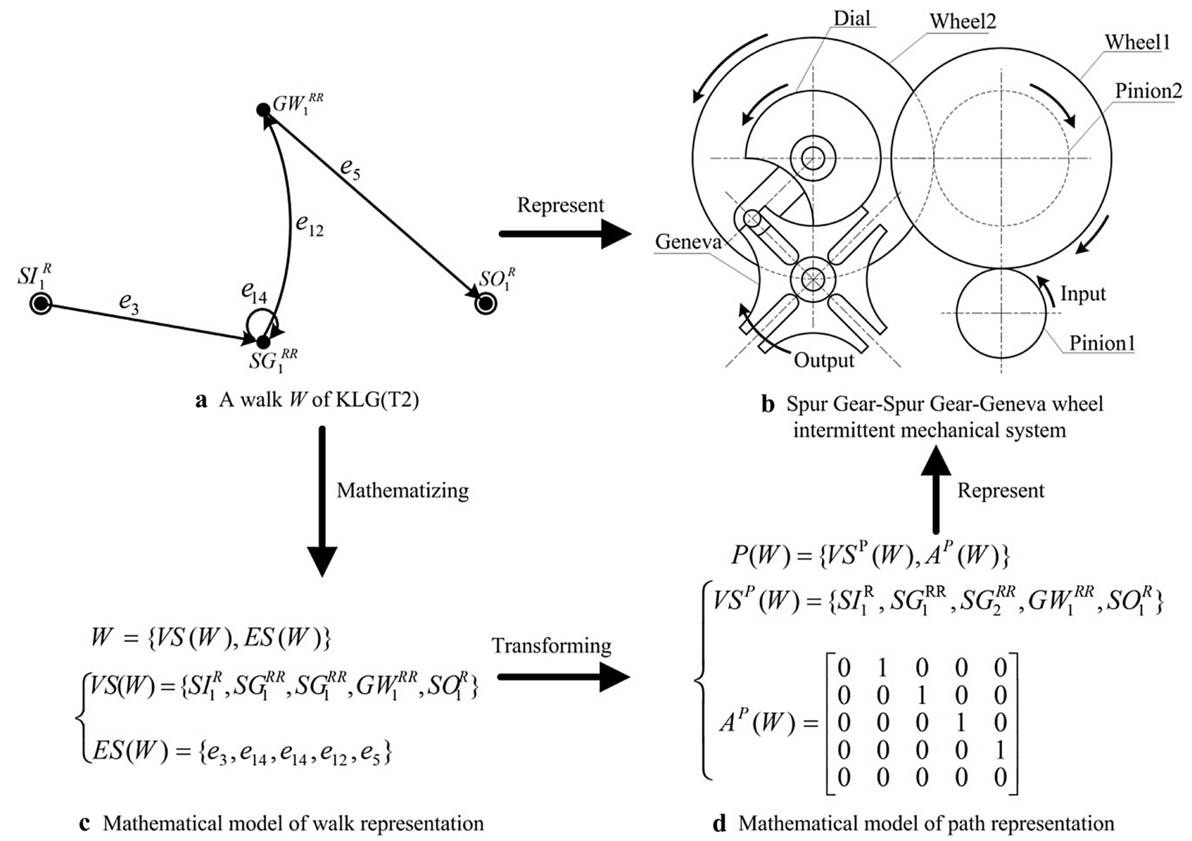
Figure 4(c) shows the VS(W) and ES(W) of the walk W in Figure 4(a). In a walk, the terms adjacent to one edge are two vertices, and these two vertices are exactly the initial vertex and terminal vertex of that edge. Thus, once the ET and ES of a walk are specified, the VT and VS of that walk are determined. Based on this, the key issues of the proposed computational synthesis approach is to compute the ET and ES of the walks whose head is input-vertex and tail is output-vertex in a KLG firstly, and then figure out their VT and VS.

3.3.2 Path Representation

The path representation is used as the final storage representation of a design candidate, and it is transformed by the walk representation. The transformation relationship between them is one-to-one. Since a path is a subgraph of the graph it belongs to, it can be described by an adjacency matrix. Therefore, the general mathematical model of path representation to a design candidate is defined as

$$P(W) = \{VS^P(W), A^P(W)\}, \tag{3}$$

where $VS^P(W)$ is the vertex sequence of the path representation and it is generated by recoding the subscript


GW_1^{RR}
e_5
e_{12}
SI_1^R
e_3
e_{14}
SG_1^{RR}
SO_1^R
Represent
Dial
Wheel2
Wheel1
Pinion2
Geneva
Input
Pinion1
Output
Mathematizing
Represent
$P(W)=\{VS^P(W), A^P(W)\}$
$VS^P(W)=\{SI_1^R, SG_1^{RR}, SG_2^{RR}, GW_1^{RR}, SO_1^R\}$
$A^P(W)=\begin{bmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\0&0&0&0&0\end{bmatrix}$
$W=\{VS(W), ES(W)\}$
$VS(W)=\{SI_1^R, SG_1^{RR}, SG_1^{RR}, GW_1^{RR}, SO_1^R\}$
$ES(W)=\{e_3, e_{14}, e_{14}, e_{12}, e_5\}$
Transforming

a A walk *W* of KLG(T2)

b Spur Gear-Spur Gear-Geneva wheel intermittent mechanical system

c Mathematical model of walk representation

d Mathematical model of path representation

Figure 4 Relationship between the walk representation and path representation of a mechanical system

index of the vertex who is used more than once in the VS of corresponding walk representation. Figure 4(d) shows the path representation of Figure 4(c). Here, the vertex SG_1^{RR} is used twice in VS(W), so the second SG_1^{RR} is recoded into SG_2^{RR} in $VS^P(W)$. $A^P(W)$ is the adjacency matrix of the path representation. This matrix representation is very helpful to analysis, optimize and furthermore to find out the optimal solution in the design candidates generated through the synthesis process in the future research. That is why the path representation is more suitable than the walk representation to be the final storage representation.

3.4 Weight Matrix Theorem

The weighted matrix theorem is a theorem to determine the ET of a walk with the aid of weighted graph and weighted matrix. In graph theory, the weighted graph, which exerts a weighted value on each edge of it, has a wide application in many practical problems. The forms of weighted value are arbitrary according to the practical problem, such as numerals, functions, symbols, etc. The weighted matrix $A_\omega = [a_{ij}]$ is an extension of adjacency matrix to describe the weighted graph in a matrix way. Here, the elements a_{ij} are the weighted values of the corresponding edges.

Let edge labels $e_i (i = 1, 2, \ldots, 16)$ be the weighted values of $KLG(T2)$ in Figure 3. Then, $KLG(T2)$ becomes a weighted graph and its weighted matrix is

$$A_\omega(KLG(T2)) = \begin{bmatrix} 0 & e_1 & e_2 & 0 & e_3 & 0 & e_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e_5 \\ 0 & 0 & 0 & e_6 & 0 & e_7 & 0 & 0 \\ 0 & e_8 & e_9 & 0 & e_{10} & 0 & e_{11} & 0 \\ 0 & e_{12} & e_{13} & 0 & e_{14} & 0 & e_{15} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & e_{16} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

It is an 8×8 symbolic matrix, where its rows and columns are associated with the vertices of $KLG(T2)$, e.g., the first row and first column are associated with the vertex SI_1^R, while the last (8th) row and last column are associated with the vertex SO_1^R. As mentioned above, the adjacency matrix theorem gives a way to determine the number of walks of length k from one vertex to another in a directed graph. Based on it, a weighted matrix theorem is defined as follows.

Weighted Matrix Theorem: The edge terms(ET) of walks of length k from vertex V_i to vertex V_j in a weighted graph G_w, who regards the edge labels as the weighted values of each edge, are given by the (i, j) element of $A_{\omega}^{k}(G_w)$.

Let $W(k,V_i,V_j)$ be the walks of length k from V_i to V_j in G_w. Based on mathematical inductive method, the theorem can be proved as follows:

Proofs When $k=1$, the edge term of $W(1,V_i,V_j)$ is the edge $e(V_i,V_j)$. Thus, the theorem is definitely correct according to the definition of weighted matrix. Let $a_{i,j}^{k-1}$ and $a_{i,j}^{k}$ be the (i, j) elements of $A_{\omega}^{k-1}(G_w)$ and $A_{\omega}^{k}(G_w)$ respectively. Assuming that the edge terms (ET) of $W(k-1,V_i,V_j)$ are given by $a_{i,j}^{k-1}$. Since $A_{\omega}^{k}(G_w) = A_{\omega}^{k-1}(G_w)A_{\omega}$,

$$a_{i,j}^{k} = \left(\sum_{m=1}^{N_v} a_{i,m}^{k-1}\right) \cdot a_{m,j}, \tag{4}$$

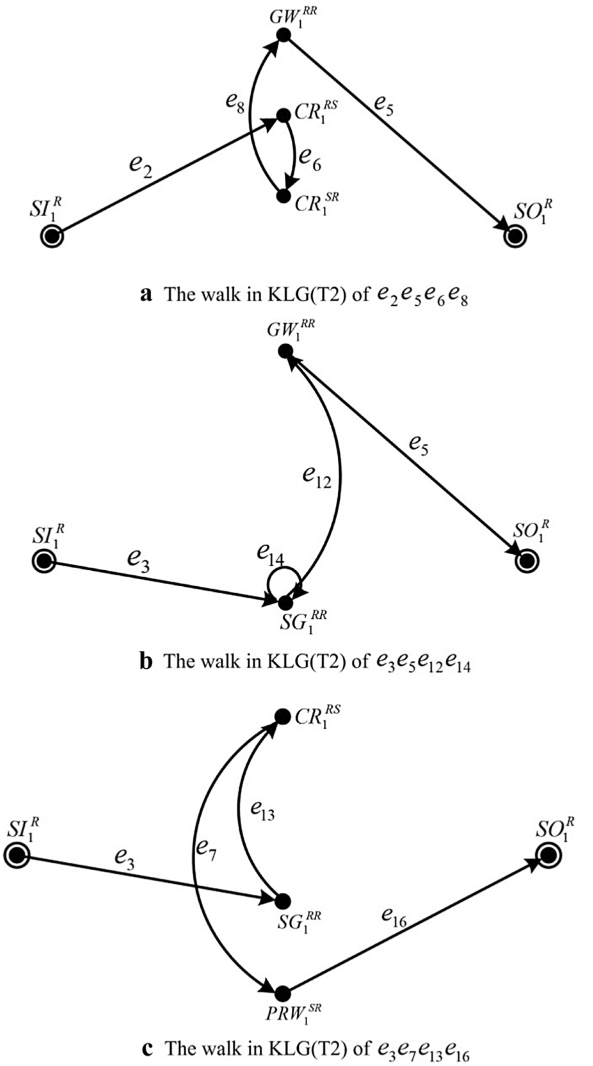
where N_v is the dimension of the weighted matrix and equals to the number of vertices of G_w. As we known, each $W(k,V_i,V_j)$ is constructed by connecting an edge $e(V_m,V_j)$ with $W(k-1,V_i,V_m)$. In Eq. (4), the value of $a_{m,j}$ is 0 or the edge label of $e(V_m,V_j)$, while the value of $a_{i,m}^{k-1}$ is 0 or a polynomial that made up by one or several monomials. Based on the assumption, $\sum_{m=1}^{N_v} a_{i,m}^{k-1}$ is the ET to all the walks of $W(k-1,V_i,V_m)$. Thus, Eq. (4) can be looked as the polynomial operation to construct $W(k,V_i,V_j)$, and $a_{i,j}^{k}$ is the ET to all the walks of $W(k,V_i,V_j)$.

In Eq. (4), the value of $a_{i,j}^{k}$ is also 0 or a polynomial. When $a_{i,j}^{k}$ is 0, there is not a walk of length k from V_i to V_j in the weighted graph. When $a_{i,j}^{k}$ is a polynomial, the alphabetic items of each monomial of it are the ET of a walk of $W(k,V_i,V_j)$ in the weighted graph. For example, the (1,8) element of $A_{\omega}^{4}(KLG(T2))$ is

$$a_{1,8}^{4} = e_2e_5e_6e_8 + e_3e_5e_{12}e_{14} + e_3e_7e_{13}e_{16}. \tag{5}$$

It can be found that $a_{1,8}^{4}$ is made up by three different monomials, and the alphabetic items of each monomial are the ET of a walk of $W(4,V_1,V_8)$ in $KLG(T2)$ as Figure 5 shows. Here, $V_1=SI_1^R$ and $V_8=SO_1^R$ as $A_{\omega}(KLG(T2))$ shows.

As mentioned above, each walk in $KLG(T2)$ whose head is *input-vertex* (i.e., SI_1^R) and tail is *output-vertex* (i.e., SO_1^R) can be looked as a design candidate. Therefore, the three monomials in polynomial $a_{1,8}^{4}$ can be looked as three different design candidates. To highlight this kind of monomial and polynomial, they are named as *monomial-walk* and *polynomial-walk* respectively in this paper. Thus, it can be said that the synthesis process of mechanical conceptual design is transformed into polynomial operations through the weighted matrix theorem.



a The walk in KLG(T2) of $e_2e_5e_6e_8$

b The walk in KLG(T2) of $e_3e_5e_{12}e_{14}$

c The walk in KLG(T2) of $e_3e_7e_{13}e_{16}$

Figure 5 Three walks in *KLG*(*T*2) corresponding to the three monomials in $a_{1,8}^{4}$

4 Polynomial Operations

4.1 Polynomial-Walk Operation

The *polynomial-walk* operation is used to calculate the ET to all the walk representations of design candidates in a *KLG*. Based on the weighted matrix theorem, a formula is given as

$$\begin{cases} P_W^{N_{\max}}(KLG) = \sum\limits_{k=2}^{N_{\max}+1} aA_{\omega}^{k-2}(KLG)b, \\ a = A_{\omega}(KLG)(m,:), \\ b = A_{\omega}(KLG)(:,n), \end{cases} \tag{6}$$

where $P_W^{N_{\max}}(KLG)$ is the computed *polynomial-walk*, while $A_{\omega}(KLG)$ is the weighted matrix of KLG. $N_{\max}$ is a positive number that denotes the maximum number of mechanisms a mechanical system may have without considering the virtual mechanisms of "System Input" and "System Output". The value of it is given by the design specifications. Moreover, m and n are the row index or column index of *input-vertex* and *output-vertex* of $A_\omega(KLG)$ respectively. Thus, a and b are the row vector and column vector of *input-vertex* and *output-vertex* in $A_\omega(KLG)$ respectively.

For example, in the weighted matrix $A_\omega(KLG(T2))$ of $KLG(T2)$, $m=1$ and $n=8$. Then, $a=(0, e_1, e_2, 0, e_3, 0, e_4, 0)$ and $b=(0, e_5, 0, 0, 0, e_{16}, 0, 0)^{\mathrm{T}}$. Set $N_{max}=3$, the *polynomial-walk* $P_W^3[KLG(T2)]$ calculated by Eq. (7) is

$$P_W^3[KLG(T_2)] = e_1e_5 + e_{12}e_3e_5 + e_{16}e_2e_7 + e_{12}e_{14}e_3e_5 + e_{13}e_{16}e_3e_7 + e_2e_5e_6e_8. \tag{7}$$

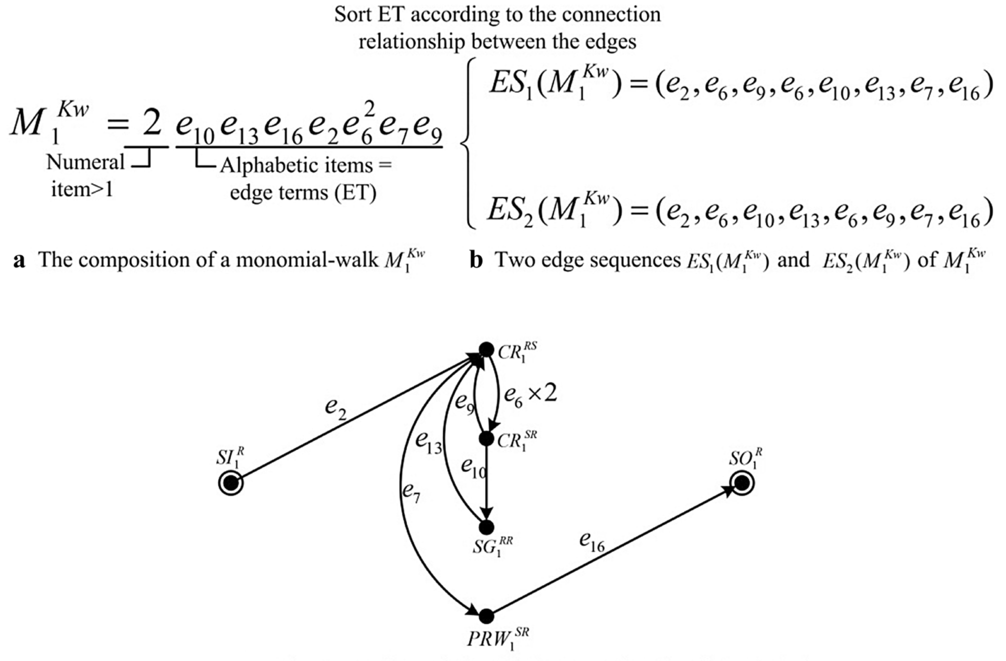
Let N_m be the number of mechanisms in a design candidate and N_e be the number of degree to the corresponding *monomial-walk*. Then, $N_m=N_e-1$. Therefore, Eq. (7) illustrates the ET to the walk representations of six feasible design candidates with the number of mechanisms of them from $N_m=1$ to $N_m=3$.

4.2 Edge Sequence Operation

Generally, every *monomial-walk* is made up by two parts, i.e., the numeral item and the alphabetic items. In the *monomial-walks* of the polynomial-walks $a_{1,8}^4$ and $P_W^3[KLG(T2)]$, the value of their numeral items are all equal to 1. However, sometimes the value of numeral item of a *monomial-walk* may be larger than 1. For example, Figure 6(a) shows the composition of M_1^{Kw}, who is a *monomial-walk* of the (1,8) element of $A_\omega^8(KLG(T2))$. Here, the value of the numeral item of M_1^{Kw} is 2. As mentioned above, the alphabetic items of a *monomial-walk* are the ET (edge terms) to the walk of a KLG. However, it should be noticed that several different walks of a KLG might have the same ET, while a walk is uniquely determined by its ES (edge sequence). It means that the ET of a *monomial-walk* may be sorted into several different ES to several different walks. Since the numeral item of a *monomial-walk* is generated by merging homogeneous monomials in the polynomial operation of Eq. (5) and Eq. (6), the value of it reveals that how many ES the ET of a *monomial-walk* can be sorted.

In Figure 6(a), the ET of M_1^{Kw} is $ET(M_1^{Kw})=\{e_{10}, e_{13}, e_{16}, e_2, e_6, e_6, e_7, e_9\}$. It can be sorted into two different edge sequences $ES_1(M_1^{Kw})$ and $ES_2(M_1^{Kw})$ (see Figure 6(b)) according to the connection relationship between the edges shown in Figure 6(c). In fact, Figure 6(c) is also the graph representation of two different walks in $KLG(T2)$ from $\mathrm{SI}_1^{\mathrm{R}}$ to $\mathrm{SO}_1^{\mathrm{R}}$. Both of these two walks have the same edge terms $ET(M_1^{Kw})$, and $ES_1(M_1^{Kw})$ and $ES_2(M_1^{Kw})$ are exactly the ES of them respectively.

Based on above analysis, the key issue of edge sequence operation is how to figure out all the edge sequences of a *monomial-walk* M^{Kw}, i.e., $ES_\ell(M^{Kw})(\ell=1,2,\ldots,N_n)$, according to its edge terms $ET(M^{Kw})$ and the connection relationship between the edges. Here, N_n is the value of numeral item of M^{Kw}.



Sort ET according to the connection relationship between the edges
$M_1^{Kw} = 2\, e_{10}e_{13}e_{16}e_2e_6^2e_7e_9$
Numeral item>1
Alphabetic items = edge terms (ET)
$ES_1(M_1^{Kw}) = (e_2, e_6, e_9, e_6, e_{10}, e_{13}, e_7, e_{16})$
$ES_2(M_1^{Kw}) = (e_2, e_6, e_{10}, e_{13}, e_6, e_9, e_7, e_{16})$
CR_1^{RS}
$e_6\times2$
e_2
e_9
CR_1^{SR}
e_{13}
SI_1^R
e_{10}
SO_1^R
e_7
SG_1^{RR}
e_{16}
PRW_1^{SR}

a The composition of a monomial-walk M_1^{Kw} **b** Two edge sequences $ES_1(M_1^{Kw})$ and $ES_2(M_1^{Kw})$ of M_1^{Kw}

c The connection relationship between the edges in KLG(T2)

Figure 6 Composition of monomial-walk $M_1^{Kw}$and its two edge sequences

Let e_i and e_j be any two edges in ET (M^{Kw}), if $ter(e_i) = init(e_j)$, then e_i and e_j may be adjacent in $ES_\ell(M^{Kw})$. Besides, let e_{in} and e_{out} be the first and last edge of $ES_\ell(M^{Kw})$ respectively, then $init(e_{in})$ equals to the *input-vertex* of KLG while $ter(e_{out})$ equals to the output-vertex of KLG. According to above statements, an edge sequence sorting algorithm is presented below.

Edge Sequence Sorting Algorithm:

```
(1): Set $k$=1, $S^{in}_{1-set} = \{\{e_{in}\}\}$ and $S^{out}_{1-set} = \{\{e_{out}\}\}$.
(2): Generate all the items $s^{in}_{ij,(k+1)-set} \in S^{in}_{(k+1)-set}$ and
     $s^{out}_{mn,(k+1)-set} \in S^{out}_{(k+1)-set}$.
{
  (2.1): $S^{in}_{(k+1)-set} = \varnothing$. For each $s^{in}_{i,k-set} \in S^{in}_{k-set}$,
         $S^{in}_{i,E} = ET(M^{Kw}) - (s^{in}_{i,k-set} \cup \{e_{out}\})$. $\forall e_j \in S^{in}_{i,E}$, if
         $Ter(s^{in}_{i,k-set}) = init(e_j)$, then
         $s^{in}_{ij,(k+1)-set} = s^{in}_{i,k-set} \cup \{e_j\}$;
  (2.2): $S^{out}_{(k+1)-set} = \varnothing$. For each $s^{out}_{m,k-set} \in S^{out}_{k-set}$,
         $S^{out}_{m,E} = ET(M^{Kw}) - (s^{out}_{m,k-set} \cup \{e_{in}\})$. $\forall e_n \in S^{out}_{m,E}$, if
         $Init(s^{out}_{m,k-set}) = ter(e_n)$, then
         $s^{out}_{mn,(k+1)-set} = s^{out}_{m,k-set} \cup \{e_n\}$;
  (2.3): Check and remove the same items in $S^{in}_{(k+1)-set}$
         and $S^{out}_{(k+1)-set}$.
}
(3): Let $N_E$ be the number of edges in $ET(M^{Kw})$. When $N_E$
     is an even number, if $k+1<N_E/2$, then $k=k+1$ and go to
     (2); Otherwise, go to (4) to find out all the $ES_\ell(M^{Kw})$,
     and then terminate. When $N_E$ is an odd number, if
     $k+1<(N_E+1)/2$, then $k=k+1$ and go to (2); Otherwise,
     go to (5) to find out all the $ES_\ell(M^{Kw})$, and then
     terminate.
(4): $\forall s^{in}_{ij,(k+1)-set}$ and $s^{out}_{mn,(k+1)-set}$, if
       $Ter(s^{in}_{ij,(k+1)-set}) = Init(s^{out}_{mn,(k+1)-set})$, then
       $s^{in}_{ij,(k+1)-set} \cup s^{out}_{mn,(k+1)-set}$ is an ES of $ES_\ell(M^{Kw})$.
(5): $\forall s^{in}_{ij,(k+1)-set}$ and $s^{out}_{mn,(k+1)-set}$, if
       $Last(s^{in}_{ij,(k+1)-set}) = First(s^{out}_{mn,(k+1)-set})$, then
       $s^{in}_{ij,(k+1)-set} \cup s^{out}_{mn,(k+1)-set}$ is an ES of $ES_\ell(M^{Kw})$.
```

Here, S^{in}_{k-set} is a set in which each item $s^{in}_{i,k-set}$ is an ordered set of k edges that starting with e_{in}. While, S^{out}_{k-set} is a set in which each item $s^{out}_{m,k-set}$ is an ordered set of k edges that ending with e_{out}. $S^{in}_{(k+)-set}$ and $S^{out}_{(k+1)-set}$ have the similar meanings with S^{in}_{k-set} and S^{out}_{k-set} respectively, instead that the items in them contain $k+1$ edges. The value of function *Ter*() equals to the terminal vertex of the last edge in corresponding ordered set, while the value of function *Init*() equals to the initial vertex of the first edge in corresponding ordered set. For example, assuming an ordered set of edges is {e_2, e_6, e_9}, then $Ter(\{e_2, e_6, e_9\}) = ter(e_9) = CR^{RS}_1$ and $Init(\{e_2, e_6, e_9\}) = init(\mathrm{e}_2) = SI^R_1$. The values of functions *First*() and *Last*() equal to the first edge and last edge in the corresponding ordered set respectively, e.g., $First(\{e_2, e_6, e_9\}) = e_2$ and $Last(\{e_2, e_6, e_9\}) = e_9$. To explain the steps of edge sequence sorting algorithm clearer, M^{Kw}_1 is taken as an example to show the execution process of the algorithm as illustrated in Table 3.

4.3 Vertex Sequence Operation

The relationship between the vertex sequence $VS_\ell(M^{Kw})$ and edge sequence $ES_\ell(M^{Kw})$ of a *monomial-walk* M^{Kw} is one-to-one. Here, the task of vertex sequence operation is to figure out all the $VS_\ell(M^{Kw})$ of M^{Kw} based on its $ES_\ell(M^{Kw})$. In order to complete this task, a formula is given as

$$VS_\ell(M^{Kw}) = [\bigcup_{i=1}^{N_E} init(e_i)] \cup ter(e_f) \quad (e_i, e_f \in ES_\ell(M^{Kw})), \tag{8}$$

where $init(e_i)$ is the initial vertex of the ith-edge e_i in $ES_\ell(M^{Kw})$, while $ter(e_f)$ is the terminal vertex of the final edge e_f in $ES_\ell(M^{Kw})$. N_E is the number of edges in $ES_\ell(M^{Kw})$. For example, $ES_1(M^{Kw}_1)$ and $ES_2(M^{Kw}_1)$ are two edge sequences of M^{Kw}_1 (see Figure 6(b)), and the final edge of both of them is e_{16}. Based on Eq. (8), the two vertex sequences $VS_1(M^{Kw}_1)$ and $VS_2(M^{Kw}_1)$ of M^{Kw}_1 are

$$\begin{cases} VS_1(M^{Kw}_1) = \{SI^R_1, CR^{RS}_1, CR^{SR}_1, CR^{RS}_1, CR^{SR}_1, SG^{RR}_1, CR^{RS}_1, \\ \qquad PRW^{SR}_1, SO^R_1\}, \\ VS_2(M^{Kw}_1) = \{SI^R_1, CR^{RS}_1, CR^{SR}_1, SG^{RR}_1, CR^{RS}_1, CR^{SR}_1, CR^{RS}_1, \\ \qquad PRW^{SR}_1, SO^R_1\}. \end{cases} \tag{9}$$

5 Computational Flowchart of Synthesis Approach

The proposed computational synthesis approach begins with the user-specified design requirements that including the input motion, output motion and the maximum number of mechanisms $N_{\max}$ in the design candidates to be generated. Then, the graph framework is built and different polynomial operations are adopted to figure out the walk representations and path representations to all the feasible design candidates. The computational flowchart of the synthesis approach is illustrated in Figure 7. To summarize and explain this figure, the steps to the computational process are:

Step 1: Set design requirements and extract KFUs from the chosen mechanisms.

Step 2: Construct KLG based on the two connection rules, and generate its vertices set $V(KLG)$ and weighted matrix $A_\omega(KLG)$ so that the KLG can be recognized and used in computer.

Table 3 Execution steps and execution results for each step of edge sequence algorithm with M_1^{Kw}acting as an example

Execution steps	Results for each step
$k=1, k+1=2<(8/2=4)$	$e_{in}=e_2 e_{out}=e_{16}$, thus $S_{1-set}^{in}=\{\{e_2\}\}, S_{1-set}^{out}=\{\{e_{16}\}\}$, then $S_{2-set}^{in}=\{\{e_2,e_6\},\{e_2,e_7\}\}$, $S_{2-set}^{out}=\{\{e_7,e_{16}\}\}$
$k=2, (k+1=3)<(8/2=4)$	$S_{3-set}^{in}=\{\{e_2,e_6,e_9\},\{e_2,e_6,e_{10}\}\}$, $S_{3-set}^{out}=\{\{e_9,e_7,e_{16}\},\{e_{13},e_7,e_{16}\}\}$
$k=3, (k+1=4)=(8/2=4)$	$S_{4-set}^{in}=\{\{e_2,e_6,e_9,e_6\},\{e_2,e_6,e_{10},e_{13}\},\{e_2,e_6,e_9,e_7\}\}$, $S_{4-set}^{out}=\{\{e_6,e_9,e_7,e_{16}\},\{e_{10},e_{13},e_7,e_{16}\}\}$. Since $Ter(\{e_2,e_6,e_9,e_6\})=Init(\{e_{10},e_{13},e_7,e_{16}\})$ $=CR_1^{SR}$, and $Ter(\{e_2,e_6,e_{10},e_{13}\})=Init(\{e_6,e_9,e_7,e_{16}\})=CR_1^{RS}$, $ES_1(M_1^{Kw})$ $=\{e_2,e_6,e_9,e_6\}\cup\{e_{10},e_{13},e_7,e_{16}\}$ $=\{e_2,e_6,e_9,e_6,e_{10},e_{13},e_7,e_{16}\}$, $ES_2(M_1^{Kw})$ $=\{e_2,e_6,e_{10},e_{13}\}\cup\{e_6,e_9,e_7,e_{16}\}$ $=\{e_2,e_6,e_{10},e_{13},e_6,e_9,e_7,e_{16}\}$

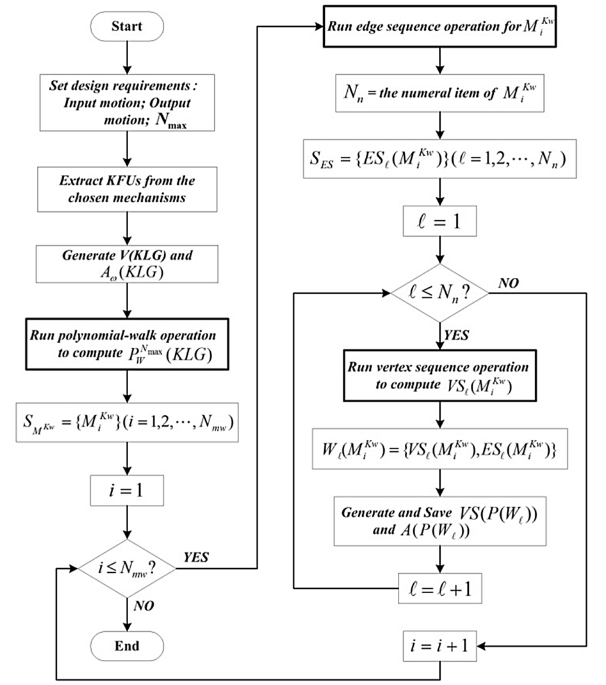

Start
Set design requirements : Input motion; Output motion; $N_{\max}$
Extract KFUs from the chosen mechanisms
Generate V(KLG) and $A_\omega(KLG)$
Run polynomial-walk operation to compute $P_W^{N_{\max}}(KLG)$
$S_{M^{Kw}}=\{M_i^{Kw}\}(i=1,2,\cdots,N_{mw})$
$i=1$
$i\le N_{mw}$?
YES
NO
End
Run edge sequence operation for M_i^{Kw}
N_n = the numeral item of M_i^{Kw}
$S_{ES}=\{ES_\ell(M_i^{Kw})\}(\ell=1,2,\cdots,N_n)$
$\ell=1$
$\ell\le N_n$?
NO
YES
Run vertex sequence operation to compute $VS_\ell(M_i^{Kw})$
$W_\ell(M_i^{Kw})=\{VS_\ell(M_i^{Kw}),ES_\ell(M_i^{Kw})\}$
Generate and Save $VS(P(W_\ell))$ and $A(P(W_\ell))$
$\ell=\ell+1$
$i=i+1$

Figure 7 Computational flowchart of the synthesis approach

Step 3: Run *polynomial-walk* operation to compute *polynomial-walk* $P_W^{N_{\max}}(KLG)$, and then separate and save all its *monomial-walks* M_i^{Kw}in set $S_{M^{Kw}}$.

Step 4: Run edge sequence operation for each M_i^{Kw} to figure out all its edge sequences $ES_\ell(M_i^{Kw})$, and then save them in set S_{ES}.

Step 5: Run vertex sequence operation for each $ES_\ell(M_i^{Kw})$ to figure out its vertex sequence $VS_\ell(M_i^{Kw})$. Then, a walk representation W_ℓ of M_i^{Kw} is formulated.

Step 6: Generated and save the vertex sequence $VS^P(W_\ell)$ and adjacency matrix $A^P(W_\ell)$ to the path representation of W_ℓ.

6 Design Example

The task here is to figure out all the feasible design candidates of a mechanical system, who is able to transform the input continuous rotational motion into the output reciprocating translational motion, from the chosen mechanisms. The maximum number of mechanisms in the mechanical system $N_{\max}$ is set to 3.

6.1 Extract and Formulate the KFUs

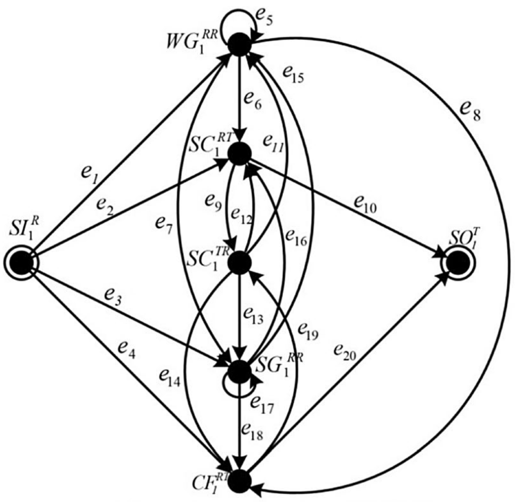
Due to space consideration of the paper, only four kinds of mechanisms, i.e., slider-crank, worm gear, cam-follower and spur gear, from the 43 identified mechanisms in Ref. [3] are chosen as the building blocks to construct the mechanical system. The KFUs extracted from these four mechanisms and two virtual mechanisms, i.e., "System Input" and "System Output", are illustrated in Table 4.

6.2 Construct Kinematic Link Graph KLG

The structure of *KLG* is up to the types of KFUs and their kinematic relationship. Based on Table 4, the graph representation to the *KLG* of design example, i.e., *KLG*(*T*4), is shown in Figure 8(a). It is a directed graph that has seven vertices and twenty edges, and the *input-vertex* is SI_1^R while the *output-vertex* is SO_1^T. The vertices set $V(KLG(T4))$ and weighted matrix $A_\omega(KLG(T4))$ of $KLG(T4)$ are shown in Figure 8(b) for subsequent computing applications.

Table 4 Seven KFUs of the design example

Mechanisms	KFUs
Slider-crank	$\{SC_1^{RT}, (R,1,-1), (T,1,1)\}$
	$\{SC_1^{TR}, (T,1,1), (R,1,-1)\}$
Spur gear	$\{SG_1^{RR}, (R,1,-1), (R,1,-1)\}$
Worm-gear	$\{WG_1^{RR}, (R,1,-1), (R,1,-1)\}$
Cam-follower	$\{CF_1^{RT}, (R,1,-1), (T,1,1)\}$
System input	$\{SI_1^{R}, (0,0,0), (R,1,-1)\}$
System output	$\{SO_1^{R}, (T,1,1), (0,0,0)\}$



a The graph representation of KLG(T4)

$$V(KLG(T4)) = \{SI_1^R, WG_1^{RR}, SC_1^{RT}, SC_1^{TR}, SG_1^{RR}, CF_1^{RT}, SO_1^T\}$$

$$A_\omega(KLG(T4)) = \begin{array}{c} \left[\begin{array}{ccccccc} 0 & e_1 & e_2 & 0 & e_3 & e_4 & 0 \\ 0 & e_5 & e_6 & 0 & e_7 & e_8 & 0 \\ 0 & 0 & 0 & e_9 & 0 & 0 & e_{10} \\ 0 & e_{11} & e_{12} & 0 & e_{13} & e_{14} & 0 \\ 0 & e_{15} & e_{16} & 0 & e_{17} & e_{18} & 0 \\ 0 & 0 & 0 & e_{19} & 0 & 0 & e_{20} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \end{array} \begin{array}{l} SI_1^R \\ WG_1^{RR} \\ SC_1^{RT} \\ SC_1^{TR} \\ SG_1^{RR} \\ CF_1^{RT} \\ SO_1^T \end{array}$$

b The vertices set and weighted matrix of KLG(T4)

Figure 8 Graph representation, vertices set and weighted matrix of *KLG*(*T*4)

6.3 Compute Polynomial-Walk

In the weighted matrix $A_\omega(KLG(T4))$, the row index of SI_1^R is m=1 while the column index of SO_1^T is n=7. Then, the row vector of SI_1^R is $a=(0,e_1,e_2,0,e_3,e_4,0)$ while the column vector of SO_1^T is $b=(0,0,e_{10},0,0,e_{20},0)^T$. Run polynomial-walk operation and substitute the values of m, n, a, b, N_{max} and $A_\omega(KLG(T4))$ into Eq. (6), the computed *polynomial-walk* $P_W^3[KLG(T4)]$ is

$$\begin{aligned} P_W^3[KLG(T4)] = {} & e_2e_{10} + e_4e_{20} + e_1e_6e_{10} + e_1e_8e_{20} \\ & + e_3e_{10}e_{16} + e_3e_{18}e_{20} + e_1e_5e_6e_{10} \\ & + e_2e_9e_{10}e_{12} + e_1e_5e_8e_{20} \\ & + e_1e_7e_{10}e_{16} + e_3e_6e_{10}e_{15} \\ & + e_2e_9e_{14}e_{20} + e_4e_{10}e_{12}e_{19} \\ & + e_1e_7e_{18}e_{20} + e_3e_8e_{15}e_{20} \\ & + e_3e_{10}e_{16}e_{17} + e_4e_{14}e_{19}e_{20} \\ & + e_3e_{17}e_{18}e_{20}. \end{aligned} \tag{10}$$

It can be found that there are 18 different *monomial-walks* $M_i^{Kw}(i=1,2,\ldots,18)$ in $P_W^3[KLG(T4)]$, whose numeral item are all equal to 1. Thus, it can be said that there are a total of 18 different design candidates that are assembled by the four different mechanisms in accordance with the design requirements.

6.4 Compute the Walk Representations and Path Representations to the Design Candidates

Run edge sequence operation and vertex sequence operation for each M_i^{Kw} in turn, the achieved edge sequences and vertex sequences of the walk representations to all design candidates are illustrated in Table 5, Column 3. To each walk representation, its path representation is transformed and illustrated in Table 5, Column 4. Furthermore, the design candidate that corresponds to the walk representation and path representation is displayed by a two-dimensional motion sketch and illustrated in Table 5, Column 5.

7 Conclusions

This paper presents a novel computable synthesis approach of mechanical conceptual design. It employs the walk and path mathematical structures in graph theory to represent the design candidates. A kinematic link graph (KLG) is constructed by looking every kinematic function unit (KFU) extracted from the mechanisms as a vertex. Two connection rules are defined to build the edges between those vertices. In KLG, each walk starting from the *input-vertex* and ending at the *output-vertex* is looked as a design candidate. Since a walk is represented by its edge sequence (ES) and vertex sequence (VS), the synthesis problem of mechanisms is turned into figuring out all the specified walks' ES and VS in KLG. Thus, a weighted matrix theorem is defined and proved. Based on this theorem, a formula is induced to compute the polynomial, in which every monomial's alphabetic items are the edge terms (ET) to one or several different walks' ES. Then, an edge sequence sorting algorithm is given to sort and split the alphabetic items into one or several different ES. Furthermore, a formula is given to compute the VS of a walk based on

Table 5 The monomial-walks, walk representations and path representations to the 18 design candidates

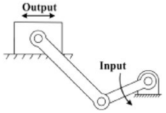
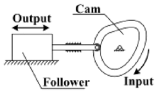
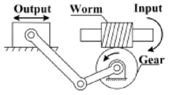
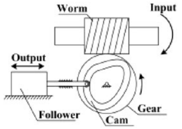
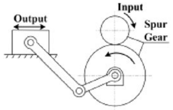
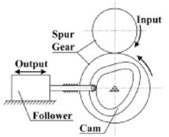
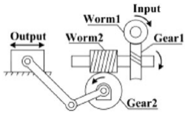
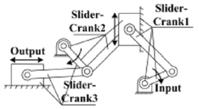
ID	Monomial-walks	Walk representations	Path representations	Design candidates
M_1^{Kw}	e_2e_{10}	$W_1(M_1^{Kw})=\begin{cases} VS_1(M_1^{Kw})=\{SI_1^R,SC_1^{RT},SO_1^T\} \\ ES_1(M_1^{Kw})=\{e_2,e_{10}\}\end{cases}$	$VS^P(W_1(M_1^{Kw}))=\{SI_1^R,SC_1^{RT},SO_1^T\}$ $A^P(W_1(M_1^{Kw}))=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}$	
M_2^{Kw}	e_4e_{20}	$W_1(M_2^{Kw})=\begin{cases} VS_1(M_2^{Kw})=\{SI_1^R,CF_1^{RT},SO_1^T\} \\ ES_1(M_2^{Kw})=\{e_4,e_{20}\}\end{cases}$	$VS^P(W_1(M_2^{Kw}))=\{SI_1^R,CF_1^{RT},SO_1^T\}$ $A^P(W_1(M_2^{Kw}))=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}$	
M_3^{Kw}	$e_1e_6e_{10}$	$W_1(M_3^{Kw})=\begin{cases} VS_1(M_3^{Kw})=\{SI_1^R,WG_1^{RR},SC_1^{RT},SO_1^T\} \\ ES_1(M_3^{Kw})=\{e_1,e_6,e_{10}\}\end{cases}$	$VS^P(W_1(M_3^{Kw}))=\{SI_1^R,WG_1^{RR},SC_1^{RT},SO_1^T\}$ $A^P(W_1(M_3^{Kw}))=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$	
M_4^{Kw}	$e_1e_8e_{20}$	$W_1(M_4^{Kw})=\begin{cases} VS_1(M_4^{Kw})=\{SI_1^R,WG_1^{RR},CF_1^{RT},SO_1^T\} \\ ES_1(M_4^{Kw})=\{e_1,e_8,e_{20}\}\end{cases}$	$VS^P(W_1(M_4^{Kw}))=\{SI_1^R,WG_1^{RR},CF_1^{RT},SO_1^T\}$ $A^P(W_1(M_4^{Kw}))=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$	
M_5^{Kw}	$e_3e_{10}e_{16}$	$W_1(M_5^{Kw})=\begin{cases} VS_1(M_5^{Kw})=\{SI_1^R,SG_1^{RR},SC_1^{RT},SO_1^T\} \\ ES_1(M_5^{Kw})=\{e_3,e_{16},e_{10}\}\end{cases}$	$VS^P(W_1(M_5^{Kw}))=\{SI_1^R,SG_1^{RR},SC_1^{RT},SO_1^T\}$ $A^P(W_1(M_5^{Kw}))=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$	
M_6^{Kw}	$e_3e_{18}e_{20}$	$W_1(M_6^{Kw})=\begin{cases} VS_1(M_6^{Kw})=\{SI_1^R,SG_1^{RR},CF_1^{RT},SO_1^T\} \\ ES_1(M_6^{Kw})=\{e_3,e_{18},e_{20}\}\end{cases}$	$VS^P(W_1(M_6^{Kw}))=\{SI_1^R,SG_1^{RR},CF_1^{RT},SO_1^T\}$ $A^P(W_1(M_6^{Kw}))=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\0&0&0&0\end{bmatrix}$	
M_7^{Kw}	$e_1e_5e_6e_{10}$	$W_1(M_7^{Kw})=\begin{cases} VS_1(M_7^{Kw})=\{SI_1^R,WG_1^{RR},WG_1^{RR},SC_1^{RT},SO_1^T\} \\ ES_1(M_7^{Kw})=\{e_1,e_5,e_6,e_{10}\}\end{cases}$	$VS^P(W_1(M_7^{Kw}))=\{SI_1^R,WG_1^{RR},WG_2^{RR},SC_1^{RT},SO_1^T\}$ $A^P(W_1(M_7^{Kw}))=\begin{bmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\0&0&0&0&0\end{bmatrix}$	
M_8^{Kw}	$e_2e_9e_{10}e_{12}$	$W_1(M_8^{Kw})=\begin{cases} VS_1(M_8^{Kw})=\{SI_1^R,SC_1^{RT},SC_1^{TR},SC_1^{RT},SO_1^T\} \\ ES_1(M_8^{Kw})=\{e_2,e_9,e_{12},e_{10}\}\end{cases}$	$VS^P(W_1(M_8^{Kw}))=\{SI_1^R,SC_1^{RT},SC_1^{TR},SC_2^{RT},SO_1^T\}$ $A^P(W_1(M_8^{Kw}))=\begin{bmatrix}0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\\0&0&0&0&0\end{bmatrix}$	 Slider- Crank2 Slider- Crank1 Output Slider- Crank3 Input

Table 5 (continued)

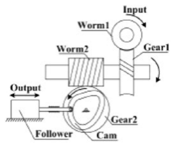
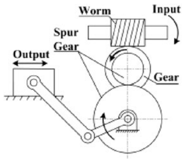
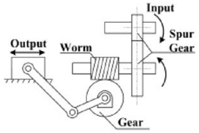
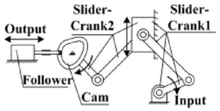
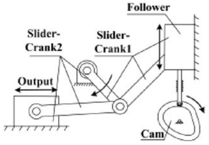
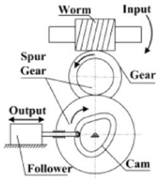
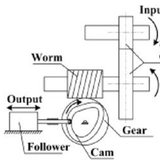
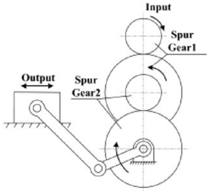
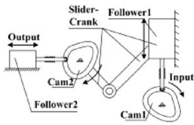
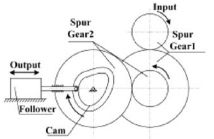
ID	Monomial-walks	Walk representations	Path representations	Design candidates
M_9^{Kw}	$e_1e_5e_8e_{20}$	$W_1(M_9^{Kw}) = \begin{cases} VS_1(M_9^{Kw}) = \{SI_1^R, WG_1^{RR}, WG_1^{RR}, CF_1^{RT}, SO_1^T\} \\ ES_1(M_9^{Kw}) = \{e_1, e_5, e_8, e_{20}\} \end{cases}$	$VS^P(W_1(M_9^{Kw})) = \{SI_1^R, WG_1^{RR}, WG_2^{RR}, CF_1^{RT}, SO_1^T\}$ $A^P(W_1(M_9^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	 Input Worm1 Worm2 Gear1 Output Gear2 Follower Cam
M_{10}^{Kw}	$e_1e_7e_{10}e_{16}$	$W_1(M_{10}^{Kw}) = \begin{cases} VS_1(M_{10}^{Kw}) = \{SI_1^R, WG_1^{RR}, SG_1^{RR}, SC_1^{RT}, SO_1^T\} \\ ES_1(M_{10}^{Kw}) = \{e_1, e_7, e_{16}, e_{10}\} \end{cases}$	$VS^P(W_1(M_{10}^{Kw})) = \{SI_1^R, WG_1^{RR}, SG_2^{RR}, SC_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{10}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	
M_{11}^{Kw}	$e_3e_6e_{10}e_{15}$	$W_1(M_{11}^{Kw}) = \begin{cases} VS_1(M_{11}^{Kw}) = \{SI_1^R, SG_1^{RR}, WG_1^{RR}, SC_1^{RT}, SO_1^T\} \\ ES_1(M_{11}^{Kw}) = \{e_3, e_{15}, e_6, e_{10}\} \end{cases}$	$VS^P(W_1(M_{11}^{Kw})) = \{SI_1^R, SG_1^{RR}, WG_2^{RR}, SC_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{11}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	
M_{12}^{Kw}	$e_2e_9e_{14}e_{20}$	$W_1(M_{12}^{Kw}) = \begin{cases} VS_1(M_{12}^{Kw}) = \{SI_1^R, SC_1^{RT}, SC_1^{TR}, CF_1^{RT}, SO_1^T\} \\ ES_1(M_{12}^{Kw}) = \{e_2, e_9, e_{14}, e_{20}\} \end{cases}$	$VS^P(W_1(M_{12}^{Kw})) = \{SI_1^R, SC_1^{RT}, SC_1^{TR}, CF_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{12}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	 Slider-Crank2 Slider-Crank1 Output Follower Cam Input
M_{13}^{Kw}	$e_4e_{10}e_{12}e_{19}$	$W_1(M_{13}^{Kw}) = \begin{cases} VS_1(M_{13}^{Kw}) = \{SI_1^R, CF_1^{RT}, SC_1^{TR}, SC_1^{RT}, SO_1^T\} \\ ES_1(M_{12}^{Kw}) = \{e_4, e_{19}, e_{12}, e_{10}\} \end{cases}$	$VS^P(W_1(M_{13}^{Kw})) = \{SI_1^R, CF_1^{RT}, SC_1^{TR}, SC_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{13}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	
M_{14}^{Kw}	$e_1e_7e_{18}e_{20}$	$W_1(M_{14}^{Kw}) = \begin{cases} VS_1(M_{14}^{Kw}) = \{SI_1^R, WG_1^{RR}, SG_1^{RR}, CF_1^{RT}, SO_1^T\} \\ ES_1(M_{14}^{Kw}) = \{e_1, e_7, e_{18}, e_{20}\} \end{cases}$	$VS^P(W_1(M_{14}^{Kw})) = \{SI_1^R, WG_1^{RR}, SG_1^{RR}, CF_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{14}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	 Worm Input Spur Gear Gear Output Follower Cam

Table 5 (continued)

ID	Monomial-walks	Walk representations	Path representations	Design candidates
M_{15}^{Kw}	$e_3e_8e_{15}e_{20}$	$W_1(M_{15}^{Kw}) = \begin{cases} VS_1(M_{15}^{Kw}) = \{SI_1^R, SG_1^{RR}, WG_1^{RR}, CF_1^{RT}, SO_1^T\} \\ ES_1(M_{15}^{Kw}) = \{e_3, e_{15}, e_8, e_{20}\} \end{cases}$	$VS^P(W_1(M_{15}^{Kw})) = \{SI_1^R, SG_1^{RR}, WG_1^{RR}, CF_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{15}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	 Input Spur Gear Worm Output Gear Follower Cam
M_{16}^{Kw}	$e_3e_{10}e_{16}e_{17}$	$W_1(M_{16}^{Kw}) = \begin{cases} VS_1(M_{16}^{Kw}) = \{SI_1^R, SG_1^{RR}, SG_1^{RR}, SC_1^{RT}, SO_1^T\} \\ ES_1(M_{16}^{Kw}) = \{e_3, e_{17}, e_{16}, e_{10}\} \end{cases}$	$VS^P(W_1(M_{16}^{Kw})) = \{SI_1^R, SG_1^{RR}, SG_2^{RR}, SC_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{16}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	
M_{17}^{Kw}	$e_4e_{14}e_{19}e_{20}$	$W_1(M_{17}^{Kw}) = \begin{cases} VS_1(M_{17}^{Kw}) = \{SI_1^R, CF_1^{RT}, SC_1^{TR}, CF_1^{RT}, SO_1^T\} \\ ES_1(M_{17}^{Kw}) = \{e_4, e_{19}, e_{14}, e_{20}\} \end{cases}$	$VS^P(W_1(M_{17}^{Kw})) = \{SI_1^R, CF_1^{RT}, SC_1^{TR}, CF_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{17}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	 Slider- Crank Follower1 Output Cam2 Input Follower2 Cam1
M_{18}^{Kw}	$e_3e_{17}e_{18}e_{20}$	$W_1(M_{18}^{Kw}) = \begin{cases} VS_1(M_{18}^{Kw}) = \{SI_1^R, SG_1^{RR}, SG_1^{RR}, CF_1^{RT}, SO_1^T\} \\ ES_1(M_{18}^{Kw}) = \{e_3, e_{17}, e_{18}, e_{20}\} \end{cases}$	$VS^P(W_1(M_{18}^{Kw})) = \{SI_1^R, SG_1^{RR}, SG_2^{RR}, CF_1^{RT}, SO_1^T\}$ $A^P(W_1(M_{18}^{Kw})) = \begin{bmatrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{bmatrix}$	 Input Spur Gear2 Spur Gear1 Output Follower Cam

its ES. The proposed synthesis approach was successfully applied in figuring out all the feasible design candidates of a mechanical system, who is able to transform the input continuous rotational motion into the output reciprocating translational motion, from the four chosen mechanisms. All the walk representations, path representations and two-dimensional motion sketches to the 18 design candidates of the design example are presented. Through the proposed synthesis approach, there is no need to search and match the mechanisms in the mechanisms library exhaustively to find out all the feasible solutions. All the feasible solutions can be figured out through one polynomial operation by the proposed approach. Besides, though the proposed approach is a graph-based approach, there is no pseudo-isomorphic graph problem in it.

Authors' contributions

GL was in charge of the whole trial; LH wrote the manuscript; XY and BH assisted with building up the framework of the research. All authors read and approved the final manuscript.

Authors' Information

Lin Han, born in 1988, is currently a PhD candidate at *Shaanxi Engineering Laboratory for Transmissions and Controls, Northwestern Polytechnical University (NWPU), China*. He received his bachelor degree from *Northwestern Polytechnical, China*, in 2011. His research interests include computer-aided design and optimization design of mechanical system.

Geng Liu, born in 1961, is currently a professor and a supervisor of PhD candidates and director of *Shaanxi Engineering Laboratory for Transmissions and Controls*, *Northwestern Polytechnical University* (*NWPU*), *China*. His research interests include mechanical transmissions; dynamics of mechanical systems; virtual and physical prototyping simulation and design technology of mechanical systems; finite element methods; contact mechanics.

Xiaohui Yang, born in 1970, is currently a professor at *Shaanxi Engineering Laboratory for Transmissions and Controls, Northwestern Polytechnical University (NWPU), China*. His research interests include mechanical design, measurement and control technology and visualization in scientific computing.

Bing Han, born in 1981, is currently an engineer at *Shaanxi Engineering Laboratory for Transmissions and Controls, Northwestern Polytechnical University (NWPU), China*. His research interests include collaborative design and simulation integration technology of mechanical system, data and process management of design process.

Competing Interests

The authors declare no competing financial interests.

Funding

Supported by State Key Program of National Natural Science Foundation of China (Grant No. 51535009), and 111 Project of China (Grant No. B13044).

Received: 12 July 2019 Revised: 6 September 2019 Accepted: 27 December 2019
Published online: 03 February 2020

References

[1] S Kota, C Lee. A functional framework for hydraulic systems design using abstraction/decomposition hierarchies. *ASME International Computers in Engineering Conference*, American Society of Mechanical Engineers, Boston, August, 1990. p. 327–340.

[2] S Kota, S J Chiou. Conceptual design of mechanisms based on computational synthesis and simulation of kinematic building blocks. *Research in Engineering Design*, 1992, 4: 75–87.

[3] S J Chiou, S Kota. Automated conceptual design of mechanisms. *Mechanism and Machine Theory*, 1999, 34: 467–495.

[4] U Yasushi, I Masaki, Y Masaharu, et al. Supporting conceptual design based on the function-behavior-state modeler. *Ai Edam Artificial Intelligent for Engineering Design Analysis & Manufacturing*, 1996, 10(4): 275–288.

[5] D Sanderson, J C Chaplin, S Ratchev. A function-behavior-structure design methodology for adaptive production systems. *International Journal of Advanced Manufacturing Technology*, 2019, 19: 1–12.

[6] H Z Zhang, X Han, R Li, et al. A new conceptual design method to support rapid and effective mapping from product design specification to concept design. *International Journal of Advanced Manufacturing Technology*, 2016, 87: 2375–2389.

[7] Y Zu, R B Xiao, X H Zhang. Automated conceptual design of mechanisms using enumeration and functional reasoning. *International Journal of Materials and Product Technology*, 2009, 34(3): 273–294.

[8] B Chen. Conceptual design synthesis based on series-parallel functional unit structure. *Journal of Engineering Design*, 2018, 29(3): 87–130.

[9] B Chen, Y B Xie. A computer-assisted automatic conceptual design system for the distributed multi-disciplinary resource environment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2016, 231(6): 1094–1112.

[10] B Chen, Y B Xie. A function unit integrating approach for the conceptual design synthesis in the distributed resource environment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2017, 232: 759–774.

[11] B Chen, Y B Xie. Functional knowledge integration of the design process. *Science China Technological Sciences*, 2016, 60(2): 209–218.

[12] J H Lee, M J Ostwald, N Gu. A syntactical and grammatical approach to architectural configuration, analysis and generation. *Architectural Science Review*, 2015, 58(3): 189–204.

[13] M I Campbell, S Kristina. Systematic rule analysis of generative design grammars. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2014, 28(3): 227–238.

[14] Y Zou, J Lü, X P Tao. Research on context of implicit context- sensitive graph grammars. *Journal of Computer Languages*, 2019, 51: 241–260.

[15] I Jowers, C Earl, G Stiny. Shapes, structures and shape grammar implementation. *Computer-Aided Design*, 2019, 111: 80–92.

[16] S Maneth, F Peternek. Grammar-based graph compression. *Information Systems*, 2018, 76: 19–45.

[17] H L Oh, T Lee, R Lipowski. A graph theory based method for functional decoupling of a design with complex interaction structure. *Proceedings of the ASME 2010 International Design Engineering Technical Conference & Computers and Information in Engineering Conference IDETC/CIE*, Montreal, Quebec, Canada, 2010: 123–132.

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

[19] C Shi, H W Guo, M Li, et al. Conceptual configuration synthesis of line-foldable type quadrangular prismatic deployable unit based on graph theory. *Mechanism and Machine Theory*, 2018, 121: 563–582.

[20] L Sun, X Chen, C Y Wu, et al. Synthesis and design of rice pot seedling transplanting mechanism based on labeled graph theory. *Computers and Electronics in Agriculture*, 2017, 143: 249–261.

[21] V V Kamesh, K M Rao, A B S Rao. An innovative approach to detect isomorphism in planar and geared kinematic chains using graph theory. *Journal of Mechanical Design*, 2017, 139(12): 122301.

[22] A Chakrabarti, K Shea, R Stone, et al. Computer-based design synthesis research: an overview. *Journal of Computing & Information Science in Engineering*, 2011, 11(2): 519–523.

[23] L Al-Hakim, A Kusiak, J Mathew. A graph-theoretic approach to conceptual design with functional perspectives. *Computer-Aided Design*, 2000, 32(14): 867–875.

[24] G Li, Z H Miao, B Li, et al. Type synthesis to design variable camber mechanisms. *Advances in Mechanical Engineering*, 2016, 8(8): 1–16.

[25] Y H Zou, P He, Y L Pei. Automatic topological structural synthesis algorithm of planar simple joint kinematic chains. *Advances in Mechanical Engineering*, 2016, 8(3): 1–12.

[26] Z F Shen, G Allison, L Cui. An integrated type and dimensional synthesis method to design one degree-of-freedom planar linkages with only revolute joints for exoskeletons. *Journal of Mechanical Design*, 2018, 140: 092302.

[27] W J Yang, H F Ding, B Zi, et al. New graph representation for planetary gear trains. *Journal of Mechanical Design*, 2018, 140: 012303.

[28] V V Kamesh, K M Rao, A B S Rao. Topological synthesis of epicyclic gear trains using vertex incidence polynomial. *Journal of Mechanical Design*, 2017, 139: 062304.

[29] B He, S Wei, Y G Wang. Computational conceptual design using space matrix. *Journal of Computing & Information Science in Engineering*, 2015, 15(1): 011004.

[30] B He, P C Zhang, L L Liu. Simultaneous functional synthesis of mechanisms with mechanical efficiency and cost. *International Journal of Advanced Manufacturing Technology*, 2014, 75: 659–665.

[31] B He, P C Zhang, J Wang. Automated synthesis of mechanisms with consideration of mechanical efficiency. *Journal of Engineering Design*, 2014, 25: 213–237.

[32] M Kobayashi, Y Suzuki, M Higashi. Integrated optimization for supporting functional and layout designs during conceptual design phase. *Proceedings of the ASME 2009 International Design Engineering Technical Conference & Computers and Information in Engineering Conference IDETC/CIE*, San Diego, California, USA, August 30-September 2, 2009: 881-889.

[33] J A Bondy, U S R Murty. *Graph theory*. Springer Berlin, 2008.