Hindawi
Scientific Programming
Volume 2022, Article ID 3601265, 12 pages
https://doi.org/10.1155/2022/3601265

Research Article

Generic Structure Construction of 3D Assembly Model Based on Conjugate Subgraph

Hu Qiao,[1] Liang You,[1] Sibo Hu,[1] Shanshan Liu,[1] and Ying Xiang[2]

[1] *School of Mechatronic Engineering, Xi'an Technological University, Xi'an 710021, Shaanxi, China*
[2] *College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an 710021, Shaanxi, China*

Correspondence should be addressed to Ying Xiang; yingcara@hotmail.com

Received 24 June 2022; Accepted 3 August 2022; Published 19 September 2022

Academic Editor: Lianhui Li

As an important resource under the background of digital design technology, 3D assembly model plays a great role in many complex product design occasions. Based on the existing 3D model database, how to find and reuse the effective information that meets their own needs in many models is one of the important ways to provide design efficiency. The research content of this paper is to build a general assembly model according to the requirements of 3D assembly model. Based on the common structure, the general structure aims to integrate as much model information as possible to meet the design needs of designers in different periods and conditions. Due to the large number of components in the 3D assembly model, blindly improving hardware cost is the root of the problem. To improve quality and efficiency from the source, we need to start with the source of input. This paper simplifies the analysis of 3D assembly model information, which can not only improve the construction efficiency of the general structure of 3D assembly model but also highlight the assembly characteristics of 3D assembly model and facilitate the matching of assembly features during the construction process. Firstly, based on the attributes of assembly features and matching relationship, combined with the idea of conjugate, the matching problem between two features in the model is solved by using the idea of conjugate subgraph matching. Secondly, based on the Ullmann algorithm, combined with the definition of conjugate subgraph and related optimization operations, a conjugate subgraph matching algorithm based on vertex screening is proposed. Finally, the construction process of the general structure is proposed, and the general structure of 3D assembly model is established.

1. Introduction

With the progress of the times and the innovation of science and technology, CAD model has made a qualitative leap in the traditional function. 3D assembly model contains much reusable information reflecting the design intention, such as structure, function, and attributes. In the design and manufacturing of products, scholars at home and abroad have done a lot of research on the mining and reuse of relevant information for CAD models [1–8]. In the process of information reuse, designers often pay more attention to local similarity comparison. Because the overall similarity almost does not exist, and the local similarity is easy to analyze and obtain, it can also better realize the reuse of model information. The construction of the general structure of 3D assembly model and the related research on design reuse are essentially the matching process of subgraph isomorphism. The process of building a general structure is mainly isomorphic matching and updating, until a comprehensive and general 3D assembly model general structure is generated. The design reuse of 3D assembly model mainly carries out subgraph isomorphic matching and semantic information matching with the general structure according to the information corresponding to the design intention provided by the designer, to provide a series of model structure information with reference value. For the NP complete problem of subgraph isomorphism, scholars at home and abroad have conducted in-depth research and achieved many results, which have been widely used in many fields.

Ullmann [9], based on the backtracking method, simplified the mapping matrix in the matching process and carried out corresponding isomorphic matching according to the adjacency relationship between vertices. Wang [10] combined the advantages of Ullmann algorithm and VF2 algorithm, first pruned through the adjacency relationship between vertices based on Ullmann algorithm. Then, based on VF2 algorithm, the vertex matching order is changed to realize fast pruning. Ma et al. [11] proposed filtering and simplification based on Ullmann algorithm. The corresponding isomorphic matching process is carried out through neighbor filtering, partial simplification, and the selection order of adding matching graph vertices. Choi et al. [12] considered various design problems and analyzed the incremental hybrid genetic algorithm for subgraph isomorphism problem to improve the performance of the algorithm. Fehér et al. [13] proposed a subgraph isomorphism algorithm based on MapReduce framework to match font patterns in any large map. Chen et al. [14] provided a method to solve the isomorphism problem of the maximum common subgraph by generating a weighted graph, in which the weight represents the probability that the associated link is in the maximum common subgraph of two input graphs. Xu et al. [15] proposed extended subgraph matching and, combined with the different characteristics of Ullmann algorithm and QuickSI, gave the corresponding edging algorithm to optimize the processing of distance information. Rong et al. [16] constructed the information representation of process model based on attribute adjacency graph and improved the subgraph isomorphism algorithm to obtain processing features. Tang [17] proposed a similarity evaluation method based on improved random walk graph matching for 3D part models and a similarity evaluation method based on tree graph matching for 3D assembly models, to improve the matching efficiency. Dong et al. [18], based on algebraic theory, taking unlabeled graph as the research object, established subgraph isomorphism algorithms from two aspects of eigenvalue construction and degree sequence, to realize the graph isomorphism matching of directed graph and undirected graph.

This paper studies the model representation method based on attribute assembly feature adjacency graph and puts forward a general structure construction method of 3D assembly model. Firstly, this method analyzes the related concepts such as assembly features and assembly relationship, introduces the idea of conjugate, and gives the definition of practical guiding significance between the matched part features. Based on the matching algorithm of conjugate subgraphs, the matching problem is solved by combining the matching algorithms of conjugate subgraphs. Finally, the construction process of general structure is formed, and the general structure of 3D assembly model is established.

2. Materials and Methods

2.1. Method for Representation of 3D Assembly Model Information. The representation of 3D assembly model information should include topological structure information that can express the structural resources related to the model, semantic information such as part name, type, and function, and some characteristic information that can express the matching relationship between parts.

Graph model representation takes graph theory as a tool and is mainly composed of nodes and edges. Graph model can be used to describe the connection relationship between structures of 3D assembly model. In the process of design and manufacturing, there are also mature methods and properties to solve relevant problems based on graph theory. Therefore, the representation method of graph model is favored by scholars and applied to many research fields.

2.1.1. Definition of Feature Adjacency Graph of Attributed Assembly

Definition (1): Attributed Adjacency Graph (AAG). It mainly takes the part face as the node and the adjacency relationship between faces as the edge of the graph representation model to represent the topological relationship of the part, which is often expressed by $G = \{V, E, \alpha, \beta\}$, wherein V represents the set of nodes, and any element v_i in the set meets the corresponding relationship with one side in the part; E represents the set of edges, which is the adjacency relationship of faces, and any element in the set has an element in V corresponding to it, mainly including the geometric type of faces, the number of edges of faces, and so on; and β represents the attribute set of edges, and any element in the set has an element in E corresponding to it, mainly including the type of edges and the position relationship of adjacent faces.

In this paper, the shape features of the model are divided into assembly features and nonassembly features. The assembly features include the assembly information in the parts, which is used to construct the main shape of the parts, which plays a decisive role in the assembly process of the 3D assembly model. Nonassembly features are auxiliary features in parts, which are local modifications of part information, and play little role in the assembly process of 3D assembly model. Based on the AAG of 3D model, this paper first maintains the assembly feature information in the part model and then uses semantic nodes to replace other information in the model, so as to construct the attributed assembly feature adjacency graph of parts. Its definition is as follows.

Definition (2): Attribute Assembly Feature Adjacency Graph. It is a graphic representation that focuses on the assembly features in the part model, which is represented by $G = \{V, E, \alpha, \beta, V_0\}$, where V represents the set of nodes, and any element v_i in the set corresponds to one side f_i in the part assembly feature; E represents the set of edges, and any element e_j in the set corresponds to the side composed of adjacent faces f_n and f_m in the part assembly feature; α represents the attribute set of the node, mainly including the geometric type of the face, the number of sides of the face, and so on; and β represents the attribute set of edges, mainly including the type of edges and the positional relationship of adjacent faces and so on. $V_0 = \{I_N, I_F, I_C\}$ represents the semantic node of the part. It is the semantic expression of other information in the part except assembly features, including semantic information such as part name (I_N), function (I_F), and category (I_C).

2.1.2. Construction of AAG of the 3D Assembly Model. From the 3D model assembly, designers can obtain comprehensive assembly information. For example, two parts in contact with each other in an assembly can be divided into contact connection and assembly connection by judging whether they have assembly properties, while assembly link represents the fact that the two parts have assembly properties and then analyzes the connection relationship. The assembly properties can be divided into riveting, key connection, pin connection, thread connection, and so on. In the process of model reuse, designers should focus on the mating connection between parts. This paper classifies and codes the connection relationship of assembly link and the contact type of mating surface, as shown in Tables 1 and 2. For example, the code "a3" represents the fact that the part assembly link relationship is riveting, and the contact type of mating surface is cylindrical-cylindrical contact.

Taking the assembly process of the 3D assembly model shown in Figure 1 as an example, the 3D assembly model shown in Figure 1(c) is assembled and combined by part A shown in Figure 1(a) and part B shown in Figure 1(b). By searching the assembly features corresponding to the model and relevant process documents, it can be obtained that the connection relationship between parts is "threaded connection," and the contact type of mating surface is "cylindrical-cylindrical contact."

Based on the construction steps of AAG of 3D assembly model, parts A and B are represented by graph models as shown in Figures 2(a) and 2(b), and the shaded part represents the mating surface. Then, code according to the contact coding rules of the mating surface above, so that the mating surfaces are assembled into a node, which is named as the mating node, that is, the node marked "d3" in the figure ("d" refers to "threaded connection" and "3" refers to "cylindrical-cylindrical contact"). Thus, the model AAG is constructed, and the results are shown in Figure 2(c).

2.2. General Structure Construction of the 3D Assembly Model Based on Conjugate Subgraph. Based on the model representation method in Section 2.1, this section realizes the construction of general structure of 3D assembly model through three steps. Firstly, the related concepts such as assembly features and fit relationship are analyzed, and the idea of conjugate is introduced to give the definition of matching part features with practical guiding significance. Then, based on the graph isomorphism algorithm and the related properties of conjugate subgraphs, the matching problem of assembly features in 3D assembly model is solved by conjugate subgraph matching algorithm based on vertex screening. Finally, the construction process of general structure is formed, and the general structure of 3D assembly model is established.

2.2.1. Relevant Definitions and Concepts

(1) Assembly Features. In the current industrial design and manufacturing, most of the features are defined as the combination of shape and function. The feature is the specific mapping of relevant shapes and is endowed with specific multisource information. In the 3D assembly model, the assembly feature is the information that represents the assembly attributes of parts; that is, it represents the assembly information such as methods and attributes contained in the cooperation between parts [19].

Table 1: Classification code of connection relationship type.

Serial number	Code	Connection relationship type
1	*a*	Riveting
2	*b*	Key connection
3	*c*	Pin connection
4	*d*	Threaded connection
5	*e*	Other

Table 2: Classification code of contact type of mating surface.

Serial number	Code	Contact type of mating surface
1	0	Plane-plane contact
2	1	Tooth-tooth contact
3	2	Plane-tooth contact
4	3	Cylindrical-cylindrical contact
5	4	Other

(2) Assembly Relationship. In the 3D assembly model, the assembly relationship plays an important role in the model expression, which can be regarded as the constraint relationship among the corresponding points, lines, and faces in the model. Analyzing the assembly relationship from the perspective of engineering design, there are many forms of assembly relationships in the actual design and manufacturing such as surface fitting, alignment, isometric surface reversing, isometric surface of the same direction, coaxial, etc., same direction isometric, coaxial and so on [20]. However, the most commonly used assembly relationships mainly include surface fitting and coaxial assembly for the consideration of assembly stability and economy.

(1) Face fitting: the two surfaces of parts in the assembly contact with opposite normal vectors. In the bolt connection shown in Figure 3, the end faces of part A and part B fit together, and the red part indicates the assembly part of surface fit.

(2) Coaxial assembly: the axes of two feature surfaces are collinear. In the bolt connection shown in Figure 3, part A and part B are coaxial with the bolt respectively, and the blue part indicates the coaxial assembly position.

(3) Conjugate Subgraph. In the actual design and manufacturing, there are many matching relationships such as those shown in Figure 3. Through the analysis of many assembly relationships, it can be observed that the topological structures of the two parts to be assembled in the 3D assembly model are usually the same, and the surface normal vector and edge attributes are often opposite. Based on the attribute assembly feature adjacency graph of parts, the idea of conjugate is introduced to express the assembly relationship between 3D assembly models. Yoke in life refers to the shelf on the back of two cows. It can play the role of allowing two cows to walk synchronously. Conjugation means that two parts are assembled with each other

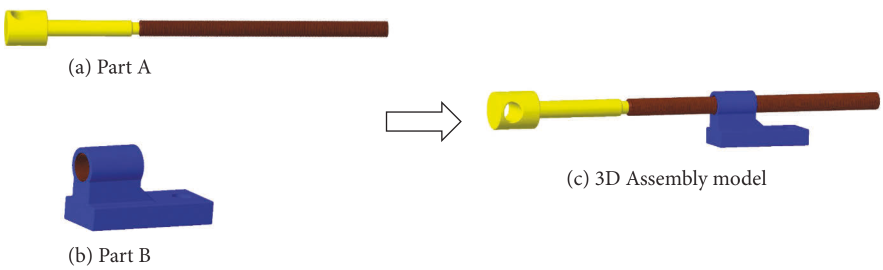


FIGURE 1: Assembly diagram of 3D assembly model. (a) Part A, (b) part B, and (c) 3D assembly model.

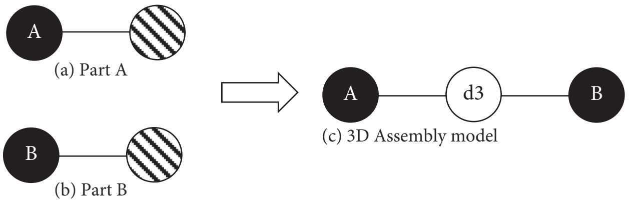


FIGURE 2: Construction diagram of AAG of the 3D assembly model. (a) Part A, (b) part B, and (c) 3D assembly model.

according to specific laws, commonly known as twinning. Twins are similar in general and slightly different in detail. Therefore, the definition of conjugate subgraph is as follows.

Definition (3): Conjugate Subgraph. In the 3D assembly model, there are two parts assembled with each other, and their attribute assembly features are adjacent to some assembly feature parts (subgraphs) represented by the graph. The two subgraphs have the same topological connection form, but the edge attributes in the graph are just opposite, so the two subgraphs are called common yoke subgraphs.

A 3D assembly model of a slider coupling is composed of Part 1, Part 2, and Part 3, shown in Figure 4. Traverse each face of the three parts of the slider coupling, and extract the geometric attributes and relevant semantic information of the fitting surface of the part assembly features as the node attributes based on the relevant definitions of the attribute assembly feature adjacency graph and conjugate subgraph. Identify the relationship between adjacent faces as the attributes of corresponding edges, so as to construct the corresponding attribute assembly feature adjacency diagram of each part, as shown in Figure 5. Among them, the two corresponding subgraphs in the green coil are conjugate subgraphs with assembly relationship between parts.

By observing the conjugate subgraph shown in Figure 5, the following properties can be obtained:

(1) Conjugate subgraphs have the same topology

(2) The face type, assembly requirements, and other attributes of each corresponding vertex in the figure are the same, but the face normal vector is opposite

(3) The attributes of the corresponding edges in the figure are opposite; for example, the concavity and convexity are opposite

(4) General Structure of the 3D Assembly Model. The common structure of the 3D assembly model is known. Each part in the common structure has several assembly features. Each part in the common structure is matched through the conjugate subgraph to match the parts and components with conjugate relationship, because, in the current assembly scene, there are often more than one component matching with the assembly features of the part. After conjugate subgraph matching, a set of AAG that meet the assembly requirements may be obtained. Therefore, a general set corresponding to the part is constructed, and its definition is shown in Definition 4.

Definition (4): General Set. The function of this set is to store the AAG of parts and components that meet the conjugate relationship with a part, which is expressed by a formula. Any graph in the set and all the adjacency graphs meet the conjugate subgraph matching with the attribute assembly feature adjacency graph of the part.

Until all the parts and components in the current scene are matched, the general set corresponding to each part meeting the conjugate relationship in the common structure is constructed, and finally a complex AAG is obtained, which is the general structure of 3D assembly model.

2.2.2. Method for Assembly Feature Matching Based on the Conjugate Subgraph. The matching problem of assembly features is basically the matching problem of conjugate subgraphs. The matching problem of conjugate subgraphs is essentially the matching problem of subgraph isomorphism. The main process includes the following steps: in the current matching environment, convert the components to be matched into the corresponding AAG representation. The conjugate subgraphs are matched with the parts and components in the common structure of the 3D assembly model, that is, the conjugate subgraphs that meet the same structure, the same node surface type, assembly requirements, and other attributes, and the opposite surface normal vector and the opposite edge attributes are searched for matching.

(1) Ullmann Algorithm. Ullmann algorithm mainly stores the matrix of the mapping relationship between the vertices of any two graphs, to judge the isomorphism of the two graphs. After one operation, this algorithm can enumerate all *isomorphic* subgraphs, so it is also called enumeration algorithm.

It is known that there are two AAG G_A and G_B, n_A and n_B represent the number of vertices of the two graphs, respectively, s_A and s_B represent the number of edges of the two

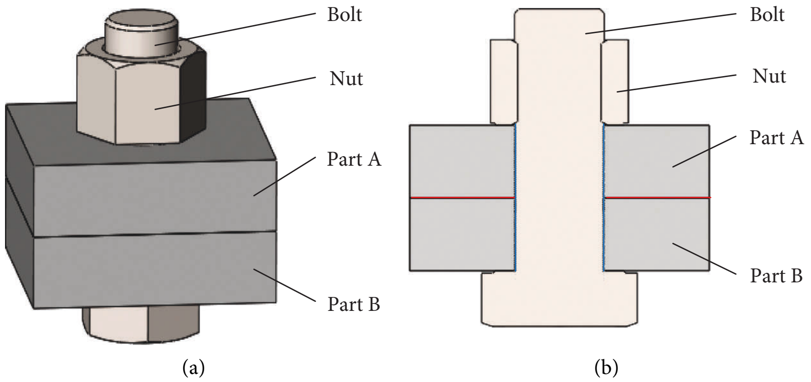


Figure 3: Bolt connection and bolt connection section. (a) Bolted connection. (b) Bolted connection section.

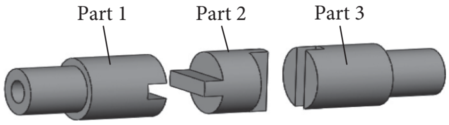


Figure 4: Slider coupling.

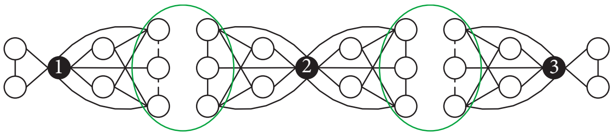


Figure 5: Attribute assembly feature adjacency diagram of slider coupling.

graphs, respectively, $n_A < n_B$, and $s_A < s_B$. $M_A = [a_{ij}]$ and $M_B = [b_{ij}]$ represent the adjacency matrix of the two graphs, respectively. The vertex correspondence between G_A and G_B is represented by vertex mapping matrix $\boldsymbol{M}$, where $\boldsymbol{M}$ is $n_A \times n_B$ matrix and m_{ij} is an element in $\boldsymbol{M}$, which mainly refers to the elements in row i and column j of the matrix, representing the matching relationship between node i of G_A and node j of G_B. If there is no mapping relationship between node i of G_A and node j of G_B, then $m_{ij} = 0$. If there is a mapping relationship between node i of G_A and node j of G_B, then $m_{ij} = 1$. If and only if each row in $\boldsymbol{M}$ has only one element $m_{ij} = 1$ and each column has at most one element $m_{ij} = 1$, it is said that the subgraphs of G_A and G_B are isomorphic.

When G_A and G_B satisfy the isomorphism condition of subgraphs, the elements in vertex mapping matrix $\boldsymbol{M}$ will remain unchanged. This is because when the element in $\boldsymbol{M}$ conforms to the formula

$$(\forall x)\big((a_{ix} = 1) \Rightarrow (\exists y)\big(m_{xy} \bullet b_{yj} = 1\big)\big) \qquad (1 \le x \le n_A, 1 \le y \le n_B). \tag{1}$$

Then, in $\boldsymbol{M}$, the elements $m_{ij} = 1$ will not change. Therefore, $\boldsymbol{M}$ constructs the vertex mapping relationship between G_A and G_B. If any two vertices in G_A have adjacency relationship, two adjacent vertices with vertex mapping relationship with G_A can also be found in G_B, so that the edge also has corresponding mapping relationship.

The basic idea of Ullmann algorithm is as follows: in the whole process of isomorphic matching, always check whether each row in $\boldsymbol{M}$ has an element $m_{ij} = 1$. If the element is not found, it means that there is no corresponding vertex in G_B that matches the vertex of G_A, and then the isomorphism condition is not tenable, so exit the program. If this element exists, then the matrix $\boldsymbol{M}$ is traversed in turn, and finally an isomorphic subgraph that meets the needs of the designer can be matched.

To sum up, the basic steps of Ullmann algorithm are as follows:

Input: G_A and G_B

Input: isomorphic subgraph.

Step 1. Initialize the vertex mapping matrix $\boldsymbol{M}$ and the incidence matrix M_0, whose size is $n_A \times n_B$.

Step 2. Define Atlas G_0 to store matching pairs of vertices.

Step 3 Starting from $r = 1$, traverse each row $r = r + 1$ in $\boldsymbol{M}$. If there is an element $m_{ij} = 1$, store the vertex matching pairs of graphs G_A and G_B corresponding to m_{ij} in G_0. The matching value of the vertex of the corresponding graph G_B in M_0 and other vertices is 0. Until $r = n_A + 1$, jump to step 6.

Step 4. Traverse all vertices in Atlas G_0; if there is a corresponding mapping relationship between the two vertices, $m_{pq} = 1$. If not, $m_{pq} = 0$.

Step 5. In the mapping matrix obtained by traversing step 4, if each row of the matrix has only one 1, and each column has at most one 1, then the subgraph of G_0 is isomorphic, and step 3 is repeated according to the matrix elements in M_0. Otherwise, the subgraph is not isomorphic, and step 3 is returned.

Step 6. At the end of the algorithm, the vertex pairs in G_0, that is, the isomorphic subgraphs of G_A and G_B, are output.

Ullmann algorithm adopts the method of depth first in matching. The whole process uses Boolean matrix to record relevant data. When there is no mapping relationship between them, it returns the nearest matching node and reexplores other matching objects. At the same

Table 3: Symbolic meaning.

Symbol	Meaning
$\boldsymbol{M}$	Mapping matrix
V_1, V_2	Vertex set: the function of the two sets is to store the matched vertices during the operation of the algorithm
m_{rc}	Elements in row r and column c in $\boldsymbol{M}$
o_c	Column C occupancy mark
B	Initialize fallback flag
L	Row count

time, while matching, check the adjacency of matching point pairs, eliminate mismatched nodes, and improve the efficiency of the algorithm. Ullmann algorithm is a mature graph isomorphism algorithm with high retrieval efficiency.

Although Ullmann algorithm is relatively mature, its retrieval efficiency is better. However, in practical engineering applications, the working time of Ullmann algorithm is exponentially related to the size of AAG. Therefore, in contrast, the whole matching process is also time-consuming. The local structure of the assembly model obtained by matching isomorphic subgraphs is also mixed. Therefore, it does not achieve better economic value and social value, and its retrieval accuracy needs to be improved accordingly. The main reasons for the poor retrieval accuracy include the following: first, in the matching process, the attributes contained in the vertices are limited and cannot be matched accurately. The consequence of fuzzy matching is the low accuracy. Secondly, isomorphic matching focuses mostly on vertices and has less constraints on edge attributes, resulting in low accuracy and efficiency.

(2) Conjugate Subgraph Matching Based on Vertex Filtering. In order to improve the retrieval accuracy of subgraph isomorphism and the efficiency of the algorithm, and to meet the needs of constructing the general structure of 3D assembly model, a conjugate subgraph matching algorithm based on vertex screening is proposed. The algorithm is mainly based on Ullmann algorithm and improves the algorithm through the following aspects:

(1) Before the algorithm starts running, the vertices of the two input graphs are filtered by using the same properties of the node attributes of the conjugate subgraph. The face type is one of the main constituent elements of the node attribute. In the 3D assembly model, if the face type is a plane, the corresponding graph should also have a plane, and if the face type is a surface, the corresponding graph should also have a surface. If there is no corresponding face type in the graph, there is no need to enter the algorithm for conjugate matching and return directly. Secondly, if the number of vertices of the subgraph is more than that of the large graph, it is impossible for the subgraph to be isomorphic and return directly. Finally, if there are a certain number of specific graphs in the subgraph, but the graph does not exist in the large graph or the number is less than the subgraph, the isomorphism cannot succeed and is returned directly. Through the above judgment, many invalid matches can be eliminated to a great extent, so as to reduce the complexity of conjugate subgraph matching algorithm and improve work efficiency.

(2) The conjugate subgraph matching model is represented based on the attribute assembly feature adjacency graph. Therefore, the attribute assembly feature adjacency graph of the model contains semantic nodes representing semantic information such as part name, category, and function, as well as matching nodes of connection relationship. In the matching, relevant information is extracted, vertex- and edge-related attributes are added, and the similarity calculation method in [21] is used to judge in the conjugate matching process, so as to make the conjugate matching achieve accurate matching, remove invalid matching as much as possible, and improve the matching accuracy.

(3) When the algorithm is in normal operation, according to the principle that the subgraphs formed by the vertex set V_A on the current small graph and V_B on the large graph must also be isomorphic, it can judge whether the subgraphs composed of V_A and V_B are isomorphic, and whether the attributes of the edges corresponding to the corresponding vertices meet the opposite properties of the corresponding edge attributes of the conjugate subgraph, so as to judge the effectiveness of the newly found columns, effectively eliminate invalid matches, and improve matching accuracy and efficiency.

In order to facilitate the description of conjugate subgraph matching algorithm, the definitions of relevant symbols in the algorithm are shown in Table 3.

To sum up, the basic steps of the algorithm are as follows.

Step 1. Pretreatment.

The specific content of preprocessing is to initialize $\boldsymbol{M}$ according to the relevant properties of conjugate subgraphs, mainly to judge the elements with the value of 1 in $\boldsymbol{M}$ to see whether 1 can be set to 0. Since the number of elements with 1 in $\boldsymbol{M}$ is also limited, the steps of preprocessing process are also limited until no element in $\boldsymbol{M}$ can be set to 0. The specific principles are as follows.

If the vertex attributes of the two graphs do not conform to the conjugate subgraph, and the corresponding node attributes of each conjugate subgraph are the same, the elements corresponding to the vertices in matrix $\boldsymbol{M}$ are set to 0.

After initialization, perform vertex filtering. If the following conditions are met, exit conjugate subgraph matching; if not, enter step 2:

(1) In $\boldsymbol{M}$, if none of the elements in a row is 1, this indicates that a node in the matched subgraph does not exist in the large graph to be retrieved.

(2) In $\boldsymbol{M}$, if $r > c$, it means that the matched subgraph is larger than the large graph to be retrieved.

(3) The number of nodes in the matched subgraph is greater than the number of similar nodes in the large graph to be retrieved.

Step 2. Initialize V_1 and V_2 so that $V_1 = 0$ and $V_2 = 0$.

Step 3. Start traversing from the first row of matrix $\boldsymbol{M}$. If element $m_{rc} = 1$, add the vertices corresponding to this element to V_1 and V_2, respectively, and set O_C as true, B as false, and L as r.

Step 4. $r + 1$, traverse matrix $\boldsymbol{M}$. If B is false, enter this line and run step 5. Otherwise, go back to line r and run step 6.

Step 5. Traverse row I in $\boldsymbol{M}$, starting from column 0:

(1) If $m_{ij} = 1$, and the column is not occupied, add the matching vertices to V_1 and V_2, respectively, and judge the conjugate subgraph matching. If the conditions such as the same conjugate subgraph structure, the same vertex face type and other attributes, the opposite face normal vector, and the opposite edge attributes are met, set O_j to true and execute step 10. Otherwise, delete the matching vertices, $L + 1$.

(2) Otherwise, $L + 1$.

Step 12. O_c in row r is false. Delete the matching vertex and traverse row I of the matrix from column $c + 1$:

(1) If $m_{ij} = 1$, and the column is not occupied, add the matching vertices to V_1 and V_2, respectively, and judge the conjugate subgraph matching. If the conditions such as the same conjugate subgraph structure, the same vertex face type and other attributes, the opposite face normal vector, and the opposite edge attributes are met, set O_j to true and B to false, and execute step 10. Otherwise, delete the matching vertex, $L + 1$.

(2) Otherwise, $L + 1$.

Step 13. If the number of columns of L and $\boldsymbol{M}$ is equal, or $c + 1$ in step 12 is equal to the number of columns of $\boldsymbol{M}$, set B to true and step back one row. Otherwise, go to the next line.

Step 14. After traversing $\boldsymbol{M}$, if V_1 and V_2 sets are not empty sets, the vertices in the set are the matched conjugate subgraphs and output. Otherwise, the algorithm ends and the subgraph is not conjugate.

To sum up, compared with Ullmann algorithm, the conjugate subgraph matching algorithm based on vertex filtering first initializes the mapping matrix based on the characteristic attributes of the conjugate subgraph in the preprocessing stage and, at the same time, carries out vertex filtering before the conjugate subgraph matching to screen and eliminate the subgraphs that do not meet the requirements, reduce the number of invalid matching processes, and improve the matching efficiency of the conjugate subgraph. Then, in the process of conjugate subgraph matching, according to the principle that the subgraphs composed of two matched sets of V_1 and V_2 must be isomorphic, combined with the relevant properties of conjugate subgraphs, judge the newly added vertices and edges and screen and eliminate the vertices that do not meet the requirements, so as to improve the matching efficiency of conjugate subgraphs.

2.2.3. Construction of General Structure. The general structure of 3D assembly model takes the common structure as the main body and combines the multisource and discrete noncommon structure information with it through the assembly feature matching algorithm based on conjugate subgraph to construct a complex and comprehensive expression form of 3D assembly model. Not only can the general structure reflect the reusable common structure information such as functions, structures, and attributes in a group of 3D assembly models, meet the needs of the public, and provide designers with the general structure of the group of models and relevant functions, attributes, design experience, and other information, but it also can reflect the relevant information of many personalized structures and include as many qualified structures as possible, so as to meet the various personalized needs of designers at different stages and under different conditions and provide comprehensive and detailed reusable information for designers in design and manufacturing.

To sum up, the construction steps of general structure of 3D assembly model are as follows:

Step 1: define n general sets (n is the number of parts in the common structure of the 3D assembly model) and initialize them to store the AAG that satisfies the conjugate matching with the part in the matching process.

Step 2: input the common structure of the 3D assembly model and the AAG Atlas corresponding to the parts and components to be matched. The Atlas mainly stores the AAG corresponding to each part and component in the model.

Step 3: based on the matching algorithm of conjugate subgraph in Section 2.2.2, traverse each part in the common structure of 3D assembly model, and match each component in the current component graph set with the AAG of the current part in turn. For the k-th part in the common structure,

(1) If the conjugate matching is satisfied, the AAG corresponding to the component is added to the

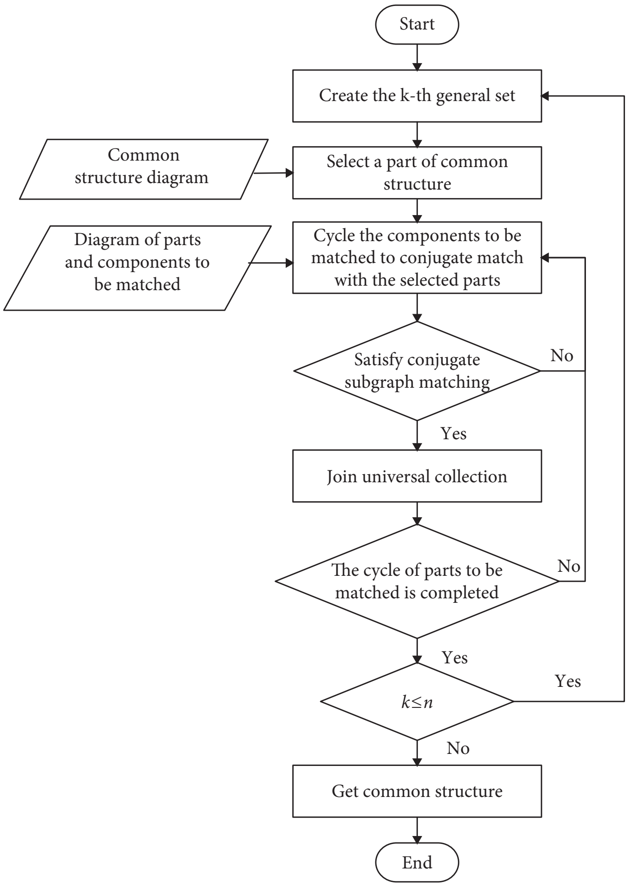

Start
Create the k-th general set
Common structure diagram
Select a part of common structure
Diagram of parts and components to be matched
Cycle the components to be matched to conjugate match with the selected parts
Satisfy conjugate subgraph matching
No
Yes
Join universal collection
The cycle of parts to be matched is completed
No
Yes
$k \leq n$
Yes
No
Get common structure
End

FIGURE 6: General structure construction flowchart of the 3D assembly model.

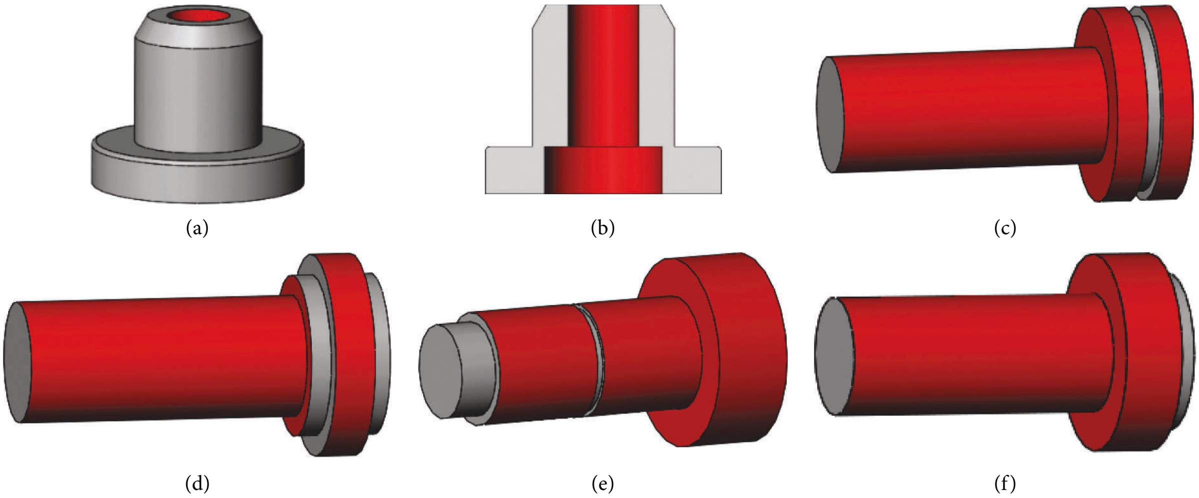


FIGURE 7: Hydraulic piston cylinder and piston rod. (a) Piston cylinder. (b) Piston cylinder section. (c) Piston rod 1. (d) Piston rod 2. (e) Piston rod 3. (f) Piston rod 4.

general set corresponding to the k-th part, the general structure state of the added 3D assembly model is updated, and the next component to be matched is selected to continue to judge the conjugate subgraph;

(2) If the conjugate matching is not satisfied, skip directly, and select the next component to be matched to continue to judge the conjugate subgraph.

Step 4: until all components in the current component diagram set to be matched are traversed, the general set

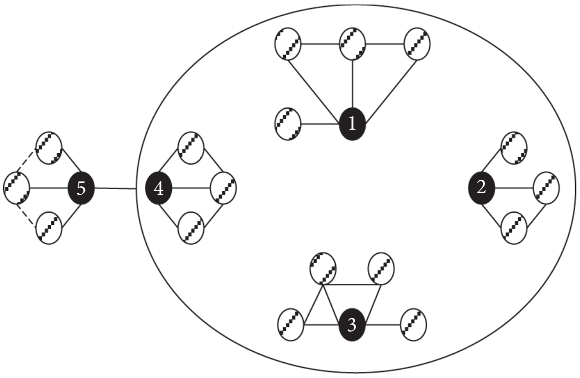


FIGURE 8: Example of general set of 3D assembly model. 1: piston rod 1; 2: piston rod 2; 3: piston rod 3; 4: piston rod 4; 5: piston cylinder.

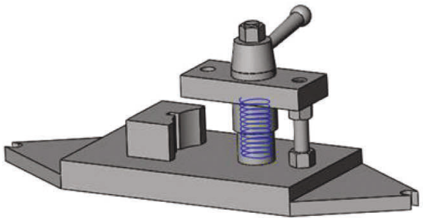


FIGURE 9: Fixture 3D assembly model.

corresponding to the part is constructed, and the part in the next common structure is selected to enter step 3.

Step 5: repeat the above steps until all parts in the common structure of the 3D assembly model are traversed, and the corresponding general set of parts is obtained, so that the general structure of the 3D assembly model can be constructed. The general assembly flowchart is shown in Figure 6.

As shown in Figure 7, there are one piston cylinder part and four piston rod parts of hydraulic transmission in machine tool accessories. Figures 7(a) and 7(b), respectively, represent the 3D model and sectional view of piston cylinder parts, parts 7(c)–7(f), respectively, represent piston rod parts of different specifications and models, and piston rod parts 1–4 meet the conjugate subgraph matching with piston cylinder parts, which can be assembled with piston cylinder parts. In the figure, the red part is the assembly feature part, and the gray part is the nonassembly feature part.

For the piston cylinder parts and piston rod parts in Figure 7, the assembly feature information is retained, and the nonassembly feature information is simplified, so as to construct the attribute assembly feature adjacency diagram corresponding to each part respectively. According to this information, the general structure is constructed, and the corresponding general set is established. The results are shown in Figure 8. The part circled by the coil is the relevant content represented by the general set. The shaded part is the assembly feature part matched between parts, and the virtual and real lines used for the connection between conjugate nodes represent the edges with opposite attributes in the assembly feature of parts.

To sum up, this method constructs the general structure of 3D assembly model based on comprehensively considering the relevant information of common structure and noncommon structure of 3D assembly model, which has a certain reference value for obtaining design commonness and mining and reuse of 3D assembly model meeting design individuality.

3. Discussion

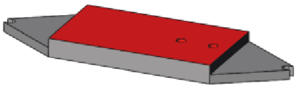
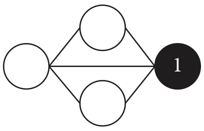
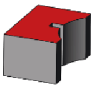
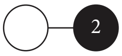
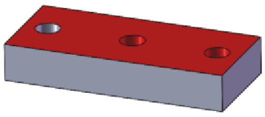
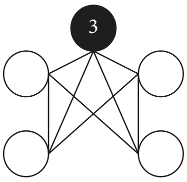
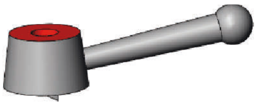
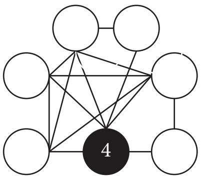
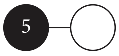
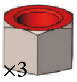
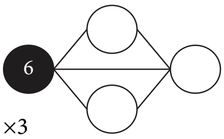
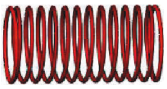
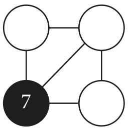
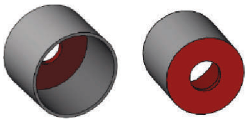
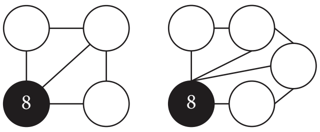
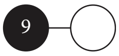
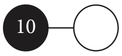
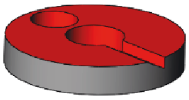
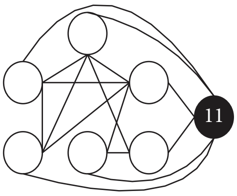
In order to verify the feasibility of the general structure construction of the 3D assembly model based on conjugate subgraph, the 3D assembly model of a machine tool fixture is selected as the research object, and the method in this paper is used to describe it in detail and step by step to verify the rationality and feasibility of the relevant methods.

Taking the 3D assembly model of a machine tool fixture shown in Figure 9 as the verification object, a detailed example verification process is carried out based on the general structure construction of conjugate subgraph.

Firstly, based on the AAG, the relevant information of assembly features in the 3D assembly model is retained, and semantic nodes are used to replace other information of parts. The corresponding attribute assembly feature adjacency graph is constructed for all parts in the model. The representation results are shown in Table 4. In the table, the red part in the part model corresponds to the assembly feature part of the part. The black node of the attribute assembly feature adjacency graph corresponds to the semantic node containing much semantic information of the part.

Then, based on the construction method of AAG of 3D assembly model, the corresponding AAG of 3D assembly model is constructed, and the representation results are shown in Figure 10.

Table 4: Attribute assembly feature adjacency diagram of each part in the fixture.

Serial number	Part name	Part model	Attribute assembly feature adjacency graph
1	Floor		
2	Positioning block		
3	Pressing plate		
4	Handle		
5	Support 1		
6	Nut		
7	Spring		
8	Spring protective sleeve		
9	Support 2		
10	Cylindrical pin		
11	Rotating parts		

The conjugate subgraph matching algorithm based on vertex screening goes through the general structure construction process in turn and then establishes a comprehensive and rich general structure expression. The experimental results are shown in Figure 11.

By analyzing the general structure of the 3D assembly model constructed in Figure 11, it can be seen that the structure displayed outside the coil is the common structure of the 3D assembly model, which represents the general composition of the model components, mainly including bottom plate, pressing plate, positioning block, support, nut, and other parts. The part circled by the coil is the general set corresponding to Part 9 (support 2) in the general structure of the 3D assembly model, that is, the noncommon structure of the 3D assembly model, which represents the product structure of different designers according to different application scenarios, functional requirements, structural optimization, and other different design concepts.

Through the above analysis, designers have a specific understanding of the general composition of this group of 3D assembly structures, and the corresponding general set contains many personalized structures to meet the design needs of different designers.

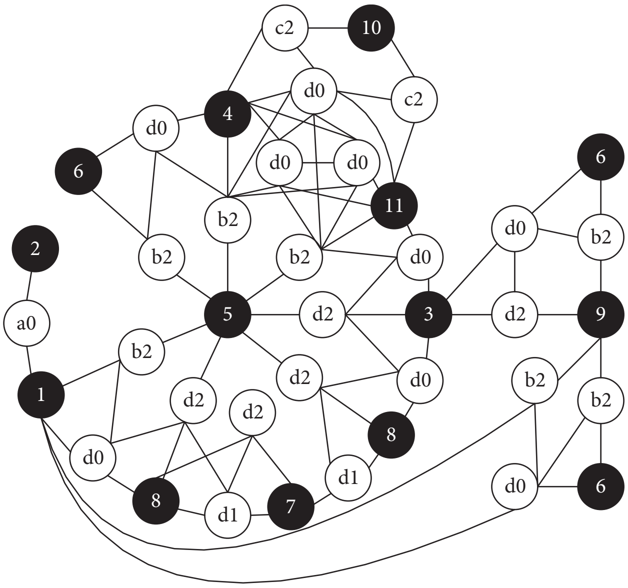


Figure 10: Fixture AAG. 1: floor; 2: positioning block; 3: pressing plate; 4: handle; 5: support 1; 6: nut; 7: spring; 8: spring protective sleeve; 9: support 2; 10: cylindrical pin; 11: rotating parts.

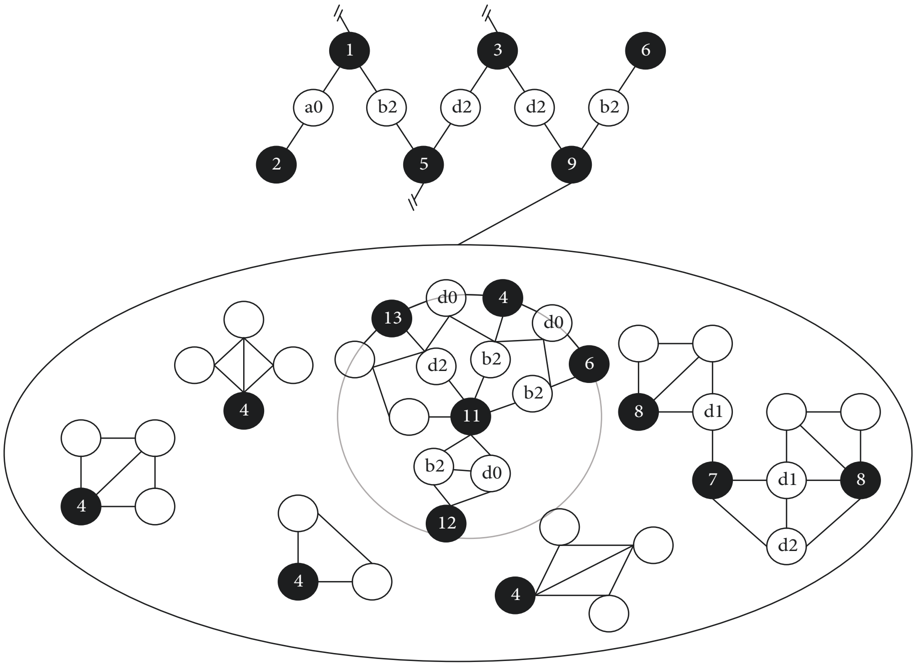


Figure 11: General structure of 3D assembly model. 1: floor; 2: positioning block; 3: pressing plate; 4: handle; 5: support 1; 6: nut; 7: spring; 8: spring protective sleeve; 9: support 2; 10: cylindrical pin; 11: rotating parts; 12: connecting rod; 13: swing rod; 14: sleeve; 15: screw.

4. Conclusion

By analyzing the concepts of assembly feature, assembly relationship, and conjugate in 3D assembly model, a general structure construction method based on conjugate subgraph is proposed in this paper. On the basis of relevant definitions and concepts, based on the relevant characteristics of conjugate subgraphs and preprocessing, the efficiency is improved, and the conjugate subgraph matching algorithm based on vertex screening is used to solve the matching problem of assembly features in 3D assembly model. On this basis, taking the common structure as the core, the construction steps of the general structure of 3D assembly model are designed, so as to build a general structure expression form with comprehensive information, reduce the matching times in the process of information retrieval, and improve the reuse efficiency of model design.

In the process of product design and manufacturing, designers have many objective and subjective requirements for the information of 3D assembly model, which have many uncertainties. Therefore, the design intention can be analyzed concretely in the follow-up, and its constituent characteristics can be summarized to build a more comprehensive semantic information expression standard that highlights the design personality.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Shaanxi Science and Technology Resources Open Sharing Platform (Grant no. 2021PT-006) and Shaanxi Science and Technology New Star Project (Grant no. 2020KJXX-071).

References

[1] K. Lupinetti, L. Chiang, F. Giannini, M. Monti, and J. P. Pernot, "Regular patterns of repeated elements in CAD assembly model retrieval," *Computer-Aided Design and Applications*, vol. 14, no. 4, pp. 516–525, 2017.

[2] G. song, *"Semantic Based Assembly Model Retrieval [D]" Hangzhou*, Springer, Berlin/Heidelberg, Germany, 2011.

[3] D. McWherter, M. Peabody, W. C. Regli, and A. Shokoufandeh, "Transformation invariant shape similarity comparison of solid models," in *Proceedings of the ASME Design Engineering Technical Conference*, pp. 46–57, Philadelphia, January 2001.

[4] Hu Qiao, Y. Zhou, and Yu Bai, "Automatic semantic annotation method of MBD model using feature recognition technology [J]," *Journal of Huaqiao University*, vol. 39, no. 05, pp. 750–755, 2018.

[5] C. L. Sun, D. Y. Ning, W. Xiong, and H. T. Wang, "Feature based CAD model retrieval technology in engineering field," *Computer integrated manufacturing system*, vol. 20, no. 04, pp. 747–754, 2014.

[6] H. Qiao, Q. Wu, S. Yu, J. Du, and Y. Xiang, "A 3D assembly model retrieval method based on assembly information," *Assembly Automation*, vol. 39, no. 4, pp. 556–565, 2019.

[7] R. Wu, *Research on Key Assembly Technologies for Process Reuse [D]" Changsha*, National University of Defense Technology, Changsha, China, 2008.

[8] T. Zheng, *"Research on 3D CAD Model Retrieval Technology [D]" Wuhan*, Huazhong University of Science and Technology, Wuhan, China, 2009.

[9] J. R. Ullmann, "An algorithm for subgraph isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, 1976.

[10] H. Wang, *"Research on Query Technology of Accurate Subgraph Database [D]" Harbin*, Harbin Institute of technology, Harbin, China, 2014.

[11] F. Yuan, *"Research and Implementation of Subgraph Isomorphism Algorithm Based on Spark [D]" Changchun: Northeast*, Normal University, Cebu, Philippines, 2018.

[12] H. G. Choi, J. Kim, Y. Yoon, and B. R. Moon, "Investigation of incremental hybrid genetic algorithm with subgraph isomorphism problem," *Swarm and Evolutionary Computation*, vol. 49, pp. 75–86, 2019.

[13] P. Fehér, M. Asztalos, T. Vajk, T. Mészáros, and L. Lengyel, "Detecting subgraph isomorphism with MapReduce," *The Journal of Supercomputing*, vol. 73, no. 5, pp. 1810–1851, 2017.

[14] A. C. L. Chen, A. Elhajj, S. Gao, A. Sarhan, A. Afra Kassem, and R. Alhajj, "Approximating the maximum common subgraph isomorphism problem with a weighted graph," *Knowledge-Based Systems*, vol. 85, pp. 265–276, 2015.

[15] Xu Kaixuan, *"Optimization Algorithm and Experimental Research of Extended Subgraph Matching Problem [D]" Shanghai*, Fudan University, Shanghai, China, 2011.

[16] Mo Rong, L. Liu, N. Wan, and J. Li, "3D process model machining feature environment matching," *Mechanical Science and Technology*, vol. 34, no. 4, pp. 549–554, 2015.

[17] W. Tang, *"Similarity Evaluation of 3D CAD Model Based on Efficient Map Matching [D]" Hangzhou*, Zhejiang University, Hangzhou, China, 2010.

[18] A. Dong, L. Gao, and J. Zhao, "Subgraph isomorphism algorithm in graph pattern mining," *Practice and understanding of mathematics*, vol. 41, no. 13, pp. 105–112, 2011.

[19] G. Wang, Y. Wang, and Z. Li, "Assembly modeling method supporting top-down die design," *China Mechanical Engineering*, no. 01, pp. 56–58, 1998.

[20] Z. Liu, *"Research on Theory, Method and Application of Product Assembly Modeling in Process and History Oriented Virtual Environment [D]" Hangzhou*, Zhejiang University, Hangzhou, China, 2002.

[21] Mi Zuo, *General Structure Mining and Information Reuse Method of 3D Assembly [D]" Xi'an*, Northwest University of technology, Seattle, Washington, 2016.