Research Article

Common Structure Mining of 3D Model Assembly Model Based on Frequent Subgraphs

Hu Qiao,1 Xufan Wang1, Zihao Wang1, Liang Wang1, and Ying Xiang2

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

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

Received 20 May 2022; Accepted 21 June 2022; Published 19 July 2022

Copyright © 2022 Hu Qiao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As an important data source of complex product design and manufacturing, 3D assembly model has accumulated a large number of 3D assembly models in various manufacturing industries under the background of widely used digital design technology. In order to make better use of reusable common structure information about 3D assembly model, reduce repetitive labor, and shorten product development cycles, a common structure mining method of 3D assembly models was proposed. Firstly, based on the attribute neighbourhood diagram of individual parts of the 3D assembly model, the information of the assembly feature attributes is maintained and the nonassembly feature attributes are simplified to form an attributed assembly feature neighbourhood diagram; then, based on the assembly relationship between the parts of the 3D assembly model, the attributed assembly feature neighbourhood diagrams of the parts are combined to form a 3D assembly model attribute neighbourhood diagram. Secondly, the common structure of the 3D assembly model is extracted by the frequent subgraph mining algorithm. Finally, a set of fixture models is used to verify the results, which show that this 3D assembly model common structure mining method can accurately and effectively explore the common structure information of the product and has good results.

1. Introduction

With the development of information science and technology, as well as the development of knowledge related to product modelling, CAD model has a qualitative leap in its ability to describe information in terms of connotation or denotation. The human being as an individual has certain three-dimensional characteristics in his or her visual senses in life, thus allowing him or her to perceive the three-dimensional model in life and its surroundings and to obtain more information.

The 3D assembly model contains many structures, functions, attributes, and other reusable information that reflect the design intent. This paper proposes a 3D assembly model common structure mining method with the 3D model as the object, as shown in Figure 1. The common parts of the graphs that are frequently found and meet the reuse requirements, namely, common structures, are then discovered. The study of the common structure discovery and reuse method will provide a reference method for the designers to discover this valuable reuse information in the design and manufacture of products and provide a reference method for the analysis and reuse of 3D assembly models.

The representation of 3D assembly model information requires comprehensive expression of relevant information on 3D assembly model so as to provide information sources for subsequent related excavation and reuse work. Therefore, the representation of 3D assembly model information should include topological structure information that can express the structural resources related to the model, semantic information that can express the name, type and function as parts, and some characteristic information that can express the coordination relationship between parts.

Chakrabarty et al. [1] proposed the hierarchical structure model of 3D assembly model for complex 3D assembly model. Liu et al. [2] specifically divided product attributes, behaviours, and other information into product layer, feature layer, and other layers, and realized the connection
relationship between information layers on the basis of the hierarchical model. Based on the representation of hierarchical model, Mou [3] constructed human-machine collaborative quantification of the mutual assembly relations between parts of 3D assembly model so as to be used in assembly sequence planning of 3D assembly model. Li [4] constructed a reuse-oriented hierarchical model, such as the inclusion relationship of structural features and the morphological relationship of features, based on the correlation topological relationship and surface hierarchical division. Yang et al. [5] proposed a two-dimensional descriptor of process information oriented to 3D assembly instruction issuing, used an abstract matrix representation method to represent 3D process information, and established an assembly instruction issuing model.

Graph model representation is mainly composed of nodes and edges based on graph theory. Graph model can be used to describe the connection relation between structure and structure of 3D assembly model. The representation method of graph model is favoured by many scholars and applied in many research fields. Bourjault [6] expressed 3D assembly model based on graph structure. The model is a simple undirected graph, whose nodes correspond to the set of parts in the model, while edges correspond to the set of assembly relations between parts. On the basis of Bourjault’s diagram model, Homem de Mello and Sanderson [7] added expressions of relevant functional attributes, thus establishing a graph model representation method of five-dimensional topology. Xiao et al. [8] proposed a single assembly interface 3D assembly model of information retrieval method: first, the preliminary model information retrieval is used, and then information is filtered according to the result of retrieval. Assembly model geometry retrieval can be converted to look for problems with the attribute adjacency graph. Finally, frequent subgraph mining algorithm is used with conjugate subgraph attribute adjacency graph search.

Zuo [9] took brake products as an example. First, the attribute connection graph expression specification is constructed, and then common design units are obtained based on clustering algorithm and frequent subgraph algorithm so as to realize assembly model information mining. Wu [10] took auto parts products as the research object, analysed the common information among products for objects at different levels, pointed out the key points to be dealt with in the reuse process, and proposed a reuse system suitable for products on this basis. Ma [11] expressed the assembly structure of products based on the number diagram of product structure and proposed the module division method of product cluster on the basis of frequent subgraph mining. Zhou et al. [12] proposed a local structure similarity analysis method of 3D assembly model based on subgraph isomorphism and case matching. Firstly, based on the attribute adjacency graph, the assembly relations between parts in 3D assembly model are transformed into corresponding elements in the graph. Secondly, the classification rules and preprocessing rules of the connection relation were defined, and similar 3D assembly model structures were matched based on the correlation algorithm. Han et al. [13] proposed a method to identify key assembly structures in complex mechanical assembly. Firstly, the model was represented by complex network, and then the key assembly functional parts were obtained based on the evaluation model. Finally, the key assembly structures were identified by heuristic algorithm. Wang et al. [14] constructed graphic descriptors to express the geometric and topological information of 3D assembly models and combined clustering algorithm and frequent subgraph mining algorithm to realize the mining of reusable models.

2. Materials and Methods

2.1. Representation of 3D Assembly Model Information. The representation of 3D assembly model information requires comprehensive expression of relevant information of 3D assembly model so as to provide information sources for subsequent related excavation and reuse work. Therefore, the representation of 3D assembly model information should include topological structure information that can express the structural resources related to the model, semantic information that can express the name, type, and function of parts, and some characteristic information that can express the coordination relationship between parts. In the representation of model information, attention should be paid to the accurate and comprehensive representation of model information. Secondly, the constructed method should be as simple, clear, and reasonable as possible, which can be applied to the subsequent related excavation and reuse work.

At present, attribute adjacency graph is used to build model, which has too much structure data and low application efficiency. Or take the whole part in the 3D assembly model as a single node and the connection relation between parts as the edge to construct the model attribute connection graph. Although these models have simple structure, they pay insufficient attention to the geometric shape information of the 3D assembly model itself. Therefore, in order to not only express important assembly information of 3D assembly model, but also improve application efficiency, this paper proposed a 3D assembly model information...
(a)

Figure 2: Continued.

(b)
2.1.1. The Definition of Attribute Adjacency Graph

Definition 1. Attribute adjacency graph: It is a graphical representation model in which the parts surface is the node and the adjacency relationship between the surfaces is the edge, so as $G = [V, E, \alpha, \beta]$ to express the topological relationship of the parts, where $V$ represents the set of nodes, and any element $v_i$ in the set meets the corresponding relationship with one side of the part; $E$ represents the set of edges, is the adjacency relation of faces; any element in the set has a corresponding element, mainly including the geometric type of faces, the number of faces, and so on. $\beta$ represents the set of attributes of an edge, and any element in the set has one element corresponding in $E$, mainly including the type of the edge and the position of the adjacent surface.

According to the current research, many scholars have proposed a variety of model representation methods. For example, shown in Figure 2(a) is the layered orientation graph model of [15] and shown in Figure 2(c) a mixture of adjacency graph model [16], respectively, the corresponding models of component assembly tolerance level, size, design, design standards, and other information in detail, and solve the problems of the corresponding field, but for the 3D assembly model related to discover. The structure is multifarious, time-consuming, and of low retrieval efficiency and high cost. The attribute connection graph model [17] as shown in Figure 2(b) is simple in form, but it ignores the geometric shape information of the model itself, resulting in low accuracy of excavation results and large differences among models.

To sum up, this paper divides the shape features of the model into assembly features and nonassembly features. Assembly features include assembly information in parts and are used to construct the body shape of parts, which plays a decisive role in the assembly process of 3D assembly models. 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. In this paper, based on the attribute adjacency graph of 3D model, the assembly feature information in the part model is preserved first, and then the semantic node is used to replace the other information in the model so as to construct the attribute assembly feature adjacency graph of the part. Its definition is as follows.

Definition 2. Property masquerade feature adjacency diagram: It is a graphical representation that focuses on assembly features in part models and 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 $f_j$ in the assembly features of parts; $E$ represents the set of edges, in which any element $e_j$ corresponds to the side formed by adjacent surfaces $f_n$ and $f_m$ in the assembly features of parts; and $\alpha$ represents the attribute set of the node, which mainly includes the geometric type of the face and the number of sides of the face. $\beta$ represents the set of edge attributes, including the type of edge and the position relation of adjacent surfaces. $V_0 = \{I_N, I_F, I_C\}$ represents the semantic node of this part, which is the semantic expression.
of other information except assembly features in this part, including the semantic information of part name ($I_N$), function ($I_F$), and category ($I_C$).

2.1.2. Construction of Attribute Adjacency Graph for 3D Assembly Model. The 3D assembly model is essentially a set of information set, that is, a collection of entities modelled by the designer in 3D space and the representation of their properties. For two parts in a 3D assembly model that are in contact with each other, two types are distinguished: contact connections and assembly connections, by determining whether they have assembly requirements. A contact connection is where two parts are in contact but there is no requirement for assembly. Conversely, where there is a requirement for assembly, the connection is an assembly connection. Secondly, assembly connections can be subdivided into welded, threaded, and pinned connections, depending on how the two parts are connected. In the process of discovering and reusing the common structure of a model, the assembly model requires special attention to the assembly relationships between parts as opposed to the part model. In this paper, the types of connection relation between parts and the contact types of mating surfaces are classified and coded, respectively, and the results are shown in Table 1 and Table 2. For example, the code “b1” means that the connection relation between two parts is “pin connection” and the contact type of the mating surface is “plane-cylinder contact.”

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

Based on the adjacency graph construction steps of attribute make-up features described above, corresponding graph representations are made for part A and part B, respectively. The results are shown in Figure 4(a) and Figure 4(b). The nodes marked with shadows in the graph represent the mating surfaces of the parts. Then, based on the classification and coding specification of the connection relation, the mating surfaces are assembled into a node, which is named as the mating node, that is, the node marked with “B2” (“B” means thread connection; “2” means cylinder-cylindrical contact). Thus, the adjacency graph of model attributes is constructed, and the result is shown in Figure 4(c).

The structure of attribute adjacency graph constructed by this method is more concise, and the graph contains the related information of part name, category, function, and so on. The size of the constructed graph is much smaller than that of the traditional method. Compared with the traditional method, the graph size is too large due to too many parts, and the complexity of the attribute adjacency graph can be reduced to a large extent, and the effect is more obvious.

2.2. Common Structure Mining of 3D Assembly Model Based on Frequent Subgraph. The excavation of the common structure of 3D assembly model is not to discover the assembly model structure with identical structure, process, and function, but to find the model structure with similar structure and different local structure in the corresponding database. In fact, the process of discovering common structures is to integrate the design experience of similar model structures and accumulate relevant design advantages so as to effectively increase the reuse efficiency of 3D assembly models.

In this paper, simplified structural data information is used as input information to reduce the scale of attribute adjacency graph of 3D assembly model and reduce the complexity of common structure excavation. In addition, the semantic nodes and coordination nodes in the attribute adjacency graph of 3D assembly model are preprocessed to reduce the size of the attribute adjacency graph of 3D assembly model.
assembly model, which is the first screening of the frequent graph mining process, before the common structure is mined through frequent subgraph. Through this process, the matching range is effectively reduced, and the overall mining efficiency is improved.

2.2.1. Related Concepts and Definitions

**Definition 3.** Graph isomorphism: Given graphs $G_A = (V_A, E_A)$ and $G_B = (V_B, E_B)$, $G_A$ and $G_B$ are isomorphic if there is a mapping between the two graphs $f: V_A \rightarrow V_B$ and $e_A = (v_A, v_A')$ is an edge in $G_A$, and if and only if $e_B = (v_B, v_B')$ is an edge in $G_B$.

**Definition 4.** Subgraph isomorphism: It is known that figures $G_A$ and $G_B$ and $G_B'$ are a subgraph of $G_B$. If there are isomorphic relationship in $G_B'$ and $G_A$, the subgraphs $G_A$ and $G_B$ are isomorphic.

The subgraph isomorphism process shown in Figure 5 is the graph isomorphism process of a subgraph of a graph and another graph. The judgment of isomorphism relation between two graphs is essentially the judgment of mapping relation between nodes in the graph, and the mapping relation under node mapping relation also satisfies the corresponding mapping relation. Among them, the left side of the two charts, respectively, as the design requirements of the model structure and database structure, the blue part is the design personnel designated by the local structure (i.e., the subgraph), two figures on the right side are, respectively, corresponding to the attribute adjacency graph, said black nodes mapping relationship of vertices, said two figures of subgraph isomorphism; namely, the dotted line represents the mapping relationship between vertices.

![Figure 5: Isomorphism diagram of subgraph.](image_url)

| Symbol | Definition |
|--------|------------|
| $V_a$  | Semantic node of part $a$ |
| $V_{a,b}$ | Mating nodes between parts $a$ and $b$ |
| $Q$    | A collection of all attribute adjacency graphs |
| $Q_P$  | All preprocessed Atlas |
| $H^K$  | Set of $K$-order candidate frequent subgraphs |
| $R^K$  | Set of frequent subgraphs of order $K$ |
| $T$    | Set of all frequent subgraphs, $T = \{R^1, R^2, \ldots, R^m\}$ |
| $id$   | In $H^K$ or $R^K$, the address number corresponding to the subgraph |
| $M$    | All vertex properties corresponding to the subgraph |
| $C_{id}$ | The adjacency matrix corresponding to the property adjacency graph |
| $C_{id}$ | In the construction of $K + 1$ candidate subgraph, it is necessary to connect two $k$-order frequent subgraphs containing the same $K$ - 1-order frequent subgraph, $C_{id}$ represents the address list of all $K - 1$-order frequent subgraphs, and $C_{id}$ is usually obtained when the candidate subgraph is generated |
| $S_{id}$ | If there is a subgraph isomorphism between a subgraph and a graph in $Q$, $S_{id}$ is used to store the address number of the graph in $Q$, and $S_{id}$ is usually obtained when frequent subgraphs are generated |
Definition 5. Frequency We know that Atlas \( Q = \{q_1, q_2, \ldots, q_n\} \) and \( q \) are subgraphs. If \( q \) appears in \( q_i \) and \( i \in n \), that is, \( q \) is isomorphic to \( q_i \) subgraph, then \( x_i \) is 1; otherwise, \( q \) is not isomorphic to any subgraph in \( q_i \); then, \( x_i \) is 0. Based on formula \( \lambda = \sum_{i=1}^{n} x_i/n \), \( \lambda \) is obtained, and \( \lambda \) is frequency \( \left( \sum_{i=1}^{n} x_i \right) \) is the sum of the number of graphs with isomorphism).

Definition 6. Frequent subgraph: The minimum frequency \( \lambda_{\text{min}} \) is known. If \( \lambda_{\text{min}} \leq \lambda \) exists, \( q \) is called its frequent subgraph in the Atlas \( Q \).

Definition 7. Generic structure: Given a 3D assembly model database, \( D = \{d_1, d_2, \ldots, d_n\} \) represents its attribute adjacency graph set, \( d \) meets the condition of frequent subgraph and is the frequent subgraph of graph \( D \). In the model database, the relevant structures of 3D assembly model corresponding to \( D \) are called common structures.

2.2.2. Discovery of Common Structure. Scholars at home and abroad often use frequent subgraph algorithm to solve common structure mining problems. Apriori algorithm has a strong influence on the mining of frequent subgraphs and is widely used [18–20]. This algorithm is a frequent item set algorithm [21, 22] that excavates and analyzes association norms. The generation of frequent item sets mainly goes through the generation of candidate sets and candidate pruning and is solved through layer-by-layer search. In other words, it generates high-order item sets on the basis of low-order item sets. In addition, the generation and testing strategies are used to discover frequent item sets, and pruning operations are carried out according to the related properties of candidate pruning to obtain high-order candidate sets. Finally, the item sets that do not meet the minimum frequency are deleted through frequency judgment so as to generate high-order frequent item sets. Two classical frequent subgraph mining algorithms, AGM algorithm [23] and FSG algorithm [24], are evolved on the basis of Apriori algorithm. AGM algorithm is based on Apriori algorithm, which generates \( k \)-order candidate set by adding one node each time. Then, the graph isomorphism is used to judge whether there are identical \( k - 1 \) candidate sets in \( k \)-order candidate sets. The algorithm needs to traverse the data set repeatedly. When the graph size increases, the running time of the algorithm will increase and the efficiency of the algorithm will decrease. The FSG algorithm generates \( k \)-order candidate sets by adding edges one at a time. In the attribute adjacency graph, there are more edges than nodes, resulting in more candidate sets. In order to optimize the efficiency of the algorithm, the intersection of the TID (Transaction ID) lists of all \( k \)-order candidate sets is calculated for frequency counting and candidate pruning. To sum up, this paper takes graph nodes as the object, combines the advantages of the two algorithms and optimization methods such as preprocessing to improve the efficiency of the algorithm, and then achieves the purpose of discovering the common structure of the model.

To facilitate the description of the algorithm, the following symbols and data structures are defined, as shown in Table 3.

The algorithm steps are described as follows:

Step 1. Pretreatment.

Step 2. Initialize the first-order and second-order frequent subgraphs.

Step 3. Connect two \( K \)-order frequent subgraphs (including the same low-order frequent subgraphs) to form \( K + 1 \)-order candidate subgraphs.

Step 4. Prune the candidate subgraph of order.

Step 5. After \( K + 1 \) candidate subgraph is generated, frequency counting is performed.

Step 6. Find and delete redundant structures in the candidate subgraph set.

Step 7. Cycle until no new frequent subset can be formed.
frequency counting, redundancy screening, and other common structure excavation processes shown in Figure 6, showing the flowchart of the algorithm. Since all elements in the attribute adjacency graph have corresponding attributes, the similarity of corresponding attributes should be judged during excavation. This paper refers to the similarity calculation method of for judgment so as to obtain more accurate excavation results.

(1) Pretreatment. The preprocessing is carried out before frequent subgraph mining, mainly by removing the semantic nodes that appear independently in each attribute adjacency graph of the graph set and the coordination nodes that are only connected with this node. That is, semantic nodes that appear only in one graph and coordination nodes that are only connected to them do not appear in other graphs. The preprocessing process is the preliminary screening of common structure excavation so as to reduce the size of Atlas and the number of matching. The following definitions of semantic node frequency are given in this chapter.

**Definition 8.** Semantic node frequency: We have \( Q = \{q_1, q_2, \ldots, q_n\} \) and \( V_a^u \). If \( V_a^u \) occurs in \( q_j \) and satisfies \( V_a^u \in q_i \), \( i \in n, j \in n, j \neq i \), then \( l_j \) is 1; otherwise, \( l_j \) is 0. The semantic node frequency \( \zeta \) is obtained by the formula \( \zeta = \sum i l_i / n \).

In Figure 7, attribute adjacency graphs corresponding to a group of 3D assembly models are used to demonstrate the
pretreatment process. Black nodes are semantic nodes, and other nodes are coordination nodes. The preprocessing mainly traverses the adjacency graph of these 3D assembly model attributes. First, the semantic node is traversed. It can be observed from the figure that only the semantic node No. 4 in Figure 7(a) exists independently, and Figure 7(b) and Figure 7(c) do not contain the semantic node, so the semantic node will be deleted during the preprocessing. Then, the coordination nodes connected with the semantic node are judged. If a coordination node is only compatible with the semantic node, the coordination node will also be deleted in the preprocessing process, as shown in Figure 7(a), and the shaded part is the coordination node only connected with the semantic node. Until the traversal of all nodes is completed, the preprocessing process is finished, and the next step is entered.

In summary, the pseudocode describing the steps of the preprocessing Algorithm 1 is as follows. Although the number of scanning Atlas increases in the pretreatment process, the number of scanning irrelevant nodes can be reduced to a large extent, the scan scale can be reduced, the matching loss can be reduced, and the excavation efficiency can be improved.

(2) Initialize. After preprocessing, the first-order frequent subgraph and second-order frequent subgraph need to be initialized. For the attribute adjacency graph of 3D assembly model, the first-order frequent subgraph can be easily obtained, which corresponds to nodes in the attribute adjacency graph, and connecting any two first-order frequent subgraphs is unique. The second-order frequent subgraph was obtained by traversing the attribute adjacency graph of

![Figure 9: Three-dimensional assembly model of fixture. (a) Tongs 1, (b) tongs 2, (c) tongs 3, (d) tongs 4, (e) tongs 5, (f) tongs 6, (g) tongs 7, and (h) tongs 8.](image-url)
the 3D assembly model. In order to facilitate the later acquisition of high-order frequent subgraphs, the id, F, M, C_{id}, and S_{id} other data structures of second-order frequent subgraphs are stored accordingly.

(3) Generation of Candidate Sets. The core idea of candidate set generation is to generate high-order candidate subgraph on the basis of low-order frequent subgraph. That is to generate K + 1-order candidate subgraph through K-order frequent subgraph. The generation of candidate set needs to go through two processes: (1) identification, judging whether two K-order frequent subgraphs contain the same K − 1-order frequent subgraphs; (2) connection: two K-order frequent subgraphs that meet the recognition conditions are connected to form K + 1-order candidate subgraphs.

The first step of candidate set generation is recognition; that is, in the corresponding two K-order frequent subgraphs (large graphs), judge whether the K − 1-order frequent subgraphs (small graphs) are the same. If they are the same, the next step can be connected. Otherwise, the connection cannot be made.

The second step in candidate set generation is joining. After identifying two identical small graphs, it is necessary to connect the large graphs containing the small graphs to obtain K + 1-order candidate subgraphs. The joining process can be understood as the joining process of adjacency matrix, namely, (K × K) + (K × K) ⇒ (K + 1) × (K + 1).

Suppose that the adjacency matrix of two identical small graphs is M^0_{K−1}, then the adjacency matrix of the two large graphs is M^a_{K} and M^b_{K}, respectively, and the matrix form is shown in formulas (1) and (2). Connect the matrices M^a_{K} and M^b_{K} so as to obtain the corresponding candidate subgraph of K + 1 order, and its matrix form is shown in formula (3).

\[
\begin{align*}
M^a_{K+1} &= \begin{bmatrix} M^0_{K−1} & a_1 \\ a_1^T & 0 \end{bmatrix}, \\
M^b_{K+1} &= \begin{bmatrix} M^0_{K−1} & b_1 \\ b_1^T & 0 \end{bmatrix}, \\
M_{K+1} &= \begin{bmatrix} M^0_{K−1} & a_1 & b_1 \\ a_1^T & 0 & c_{K,K+1} \\ b_1^T & c_{K+1,K} & 0 \end{bmatrix},
\end{align*}
\]

where \(a_1\) and \(b_1\) represent column vectors of dimension, respectively.

\[c_{K,K+1}\] represents the adjacency between \(a_1\) and \(b_1\). When the adjacency of \(a_1\) and \(b_1\) is different, the corresponding value of \(c_{K,K+1}\) is also different.

\[
\begin{align*}
c_{K,K+1} &= \begin{cases} 1, & \text{a}_1 \text{ and } b_1 \text{ have an adjacency,} \\
0, & \text{there is no adjacency between } a_1 \text{ and } b_1.\end{cases}
\end{align*}
\]

The sample graph of candidate set generation is shown in Figure 8. The two attribute adjacency graphs shown in Figure 8(a) are mainly used to carry out corresponding subgraph recognition and connection process. The two attribute adjacency graphs shown in Figure 8(a) are composed of five different vertices, respectively. In order to connect them, the same low-order frequent subgraphs need to be identified first, and the result of subgraph recognition is shown in Figure 8(b). Based on the same low-order frequent subgraph after recognition, the two attribute adjacency graphs (Figure 8(a)) are connected to obtain the high-order candidate subgraph. The process of subgraph connection is shown in Figure 8(c), where the red line box is the same low-order frequent subgraph, and the dotted line is the adjacency relationship between vertices 5 and 6. If there is adjacency relationship between the two, the dotted line is transformed into a solid line. If no adjacency exists, delete the dotted line.

After the subgraph is connected, the generated K + 1-order candidate frequent subgraph is compared with the existing graph in R^{K+1}. If there is no such K + 1-order candidate frequent subgraph in R^{K+1}, it is added to R^{K+1}, and the low-order frequent subgraph id is added to C_{id} of the high-order candidate frequent subgraph C_{id}. Otherwise, the low-order frequent subgraph id is only added to C_{id} of the high-order candidate frequent subgraph, and then the repeated data is deleted.

---

**Algorithm 1: Preprocessing algorithm.**

| Step | Action |
|------|--------|
| Step 1. | Begin |
| Step 2. | \(a = 1, n = 1\) |
| Step 3. | While \(q_\delta \neq \emptyset\) and \(q_\delta \in Q\) do \(/A\) graph in a loop Atlas |
| Step 4. | While \(V^\delta_1 \neq \emptyset\) and \(V^\delta_1 \in q_\delta\) do \(/\) Semantic nodes in a loop graph |
| Step 5. | If \(\xi(V^\delta_1) < \xi_{\min}\) then |
| Step 6. | Remove \(V^\delta_1 \cup V_{ab}\) //Remove the relevant nodes whose frequency is less than the threshold |
| Step 7. | end if |
| Step 8. | \(a = a + 1\) |
| Step 9. | end while |
| Step 10. | \(n = n + 1\) |
| Step 11. | end while |
| Step 12. | Return \(Q_p\) //Extract the preprocessed Atlas |
| Step 13. | End |
(4) **The Candidate Pruning.** In the process of frequent subgraph mining, the following operations should be performed before the frequency counting of candidate subsets generated based on the generation of candidate sets:

1. For $K+1$-order candidate subgraphs, all of their $K$-order subgraphs are solved.
2. On the basis of graph isomorphism, judge whether $K$-order frequent subgraphs contain $K$-order subgraphs.
3. According to the related properties of candidate pruning, pruning was carried out. The related properties of candidate pruning are described as follows:

   $\begin{align*}
   N < \lambda_{\text{min}} & \quad \text{Delete it and prune it,} \\
   \geq \lambda_{\text{min}} & \quad \text{Isomorphism judgment is made on the graph in the intersection with the graph.}
   \end{align*}$

The above two cases do not require a lot of subgraph isomorphism judgment, which greatly reduces the complexity of frequent subgraph algorithm and improves the retrieval efficiency of the algorithm.

(5) **Redundancy Selection.** On the basis of the Apriori algorithm, frequent subgraphs are often mined, and redundant candidate subgraphs are often generated. Since the formation of frequent subgraphs is based on the recognition and connection operation of low-order frequent subgraphs to construct high-order candidate subgraphs, the phenomenon of high-order including low-order subgraphs may occur. Therefore, this paper adopts the union of $C_{id}$ table to realize, check whether there is the same item in the union set, if there is, remove the same item, and delete the corresponding subgraph in the frequent subgraph set. This avoids scanning the whole Atlas, reduces the workload, and improves the efficiency of the algorithm.

Given subgraph $O$, if $O$ is frequent subgraph, then any subgraph $P (P \subseteq O)$ is frequent. If $O$ is infrequent, then any subgraph $S (O \subseteq S)$ is infrequent.

Through the above operations, the number of candidate subgraphs with different structures and infrequent subgraphs is reduced, and the efficiency of frequency counting process is improved to a certain extent so as to improve the efficiency of frequent subgraph mining algorithm. However, once the number of isomorphic vertices increases, the corresponding retrieval efficiency will decrease. Therefore, this paper sets the number of frequent $K$-order subgraphs in $C_{id}$ as $a$, and the number of all $K$-order subgraphs solved by $K+1$-order candidate subgraphs as $b$. By comparing $a$ and $b$, it can determine whether they meet the candidate pruning-related properties, when

In summary, the pseudocode describing the frequent subgraph mining Algorithm 2 is as follows:

Through the process of preprocessing, initialization, candidate set generation, candidate pruning, frequency counting, and redundancy screening, a large number of frequent subgraphs with different frequency were obtained. Any frequent subgraph corresponds to the local model structure frequently appearing in the model, namely, the common structure, which contains a lot of information, such as the attribute information of corresponding parts in the 3D assembly model and the assembly relationship between parts.

3. **Discussion**

In order to validate the 3D assembly model based on graph theory in common structure is found feasible, based on the laboratory model of long-term accumulation, in resources, select one series machine tool fixture for 3D assembly model...
as the research object, and use the method in this paper to
describe concrete step in detail and validate the rationality
and feasibility of relevant methods.

The 3D assembly model of a certain series of machine
tools and fixtures as shown in Figure 9 is taken as the
verification object, and the detailed example verification
process is carried out through the representation of model
information described in Section 2 and the excavation of
common structures based on frequent subgraphs.

3.1. 3D Assembly Model Information Representation.
Establishing a reasonable and effective 3D assembly model
information representation is the key to subsequent correla-
tion analysis. Therefore, based on the model representation
method described in Chapter 2, the 3D fixture assembly model
in Figure 9 is represented by an attribute adjacency graph.

3.1.1. Construction of Adjacency Graph of Attribute Con-
figuration Feature. Firstly, according to the method de-
scribed in Section 2.1.2, the relevant information of
assembly features in the 3D assembly model is retained
based on the attribute adjacency graph, and other infor-
mation of parts is replaced by semantic nodes to construct
the corresponding attribute assembly feature adjacency
diagram for all parts in the model. Fixture 2 as shown in
Figure 9(b) is used in this paper for illustration. Relevant
attribute information of parts can be obtained by searching
for model structure tree, relevant design documents, na-
tional standards, industry standards, and other resources,
and then the attribute assembly feature adjacency graph is
expressed for all parts corresponding to Fixture 2. The
results are shown in Table 4. In the table, the red part of
the part model corresponds to the assembly feature part of
the part. The black nodes in the attribute configuration adjac-
cency graph correspond to the semantic nodes containing a
lot of semantic information of the part.

3.1.2. Representation of 3D Assembly Model Information.
Through the above steps, all parts in fixture 2 can get the
corresponding attribute configuration feature adjacency
diagram. Then, based on the attribute adjacency graph

Figure 11: $\lambda_{\text{min}} = 1$ corresponds to frequent subgraph and 3D assembly model. (a) Frequent subgraph. (b) Three-dimensional assembly model.

Figure 12: $\lambda_{\text{min}} = 0.75$ corresponds to frequent subgraph and 3D assembly model. (a) Frequent subgraph. (b) Three-dimensional assembly model.
construction method in Section 2.1.3 and the second jig’s 3D assembly model shown in Figure 9(b), the assembly relations between parts are found by searching the corresponding features and relevant documents available of the jig model. Next, this section combines the assembly relations to build the corresponding 3D assembly model attribute adjacency graph. The results are shown in Figure 10(b).

Through the above steps, the 3D assembly model attribute adjacency diagram of fixture 2 is created. By using Figure 10(b), it can be visually observed that the attribute adjacency diagram contains information related to the assembly features of each part, as well as semantic nodes containing a lot of semantic information about the part, and that the diagram representation scale is greatly reduced, suggesting an information representation idea that can be used for related design and manufacturing work. Similarly, by using the same method of constructing the attribute adjacency diagram for the 3D assembly model of fixture 2, a corresponding model representation of the other machine tool fixture models in Figure 9 was made, and the results are shown in Figure 10.

3.2. The Discovery of Common Structure. Based on the attribute adjacency graph of the machine tool fixture constructed above, the method described in Section 2.2.2 is adopted to explore the common structure of the fixture model. The common structure of 3D assembly models to vary from set frequency threshold, and the excavation results from any threshold are corresponding to the corresponding common structure. In the process of verification, in order to reflect the corresponding common structure of under different thresholds, as well as the actual situation and requirements, the minimum frequency threshold ($\lambda_{\text{min}}$) is set to 1 and 0.75, respectively, in this paper. The two sets of thresholds are used to illustrate the corresponding experimental results as shown below.

When the frequency threshold is minimum (i.e., the common structures obtained by excavation are corresponding to all fixture models in this group), the frequency subgraph obtained under this threshold and the corresponding model structure of this frequency subgraph are shown in Figure 11.

Through observation and analysis of Figure 11, the basic composition of fixture model components in the common structure corresponding to the minimum frequency threshold $\lambda_{\text{min}} = 1$ can be obtained, and the model structure of this series of machine tool fixtures can be preliminarily recognized and understood. Corresponding to the examples given in this paper, the common structure of this group of fixture models under the threshold value should at least include the pressure plate, bottom plate, support parts, nuts, positioning blocks, and other components. Parts are connected by welding, thread connection, contact connection, and so on.

When the minimum frequency threshold is $\lambda_{\text{min}} = 0.75$ (i.e., in this group of models, the common structures obtained through excavation correspond to at least 6 fixture models), the frequent subgraph obtained under this threshold and the corresponding model structure of the frequent subgraph are shown in Figure 12.

Through observation and analysis of Figure 12, the common structure corresponding to the minimum frequency threshold $\lambda_{\text{min}} = 0.75$ of this group of fixture models can be obtained, which should at least include a press plate, bottom plate, support parts, spring, spring protective sleeve, nut, positioning block, and other parts. Parts are connected by welding, thread connection, contact connection, and so on.

To sum up, taking the attribute information and the assembly relation between all parts, this method can obtain the common structures which frequently appear and support the reuse demand. This method has certain reference value for the assembly relationship and parts in 3D assembly model topology similar common structure discovery and reuse. In the design and manufacture of products, designers

| Algorithm 2: The frequent subgraph mining algorithm. |
|---------------------------------------------------|
| input: $Q$ and $\lambda_{\text{min}}$          |
| Output: $T$                                   |
| Step 1. Begin                                 |
| Step 2. Pretreatment                          |
| Step 3. Initialization $H_1$, $H_2$, $K = 2$  |
| Step 4. While $H^K \neq \emptyset$ do        |
| Step 5. $H^{K+1} \leftarrow \emptyset$       |
| Step 6. $R^{K+1} \leftarrow \text{candidate}(H^K)$ //form candidate subgraphs |
| Step 7. For $q^{K+1} \in R^{K+1}$ do //candidate pruning and frequency count |
| Step 8. If $\lambda(q^{K+1}) < \lambda_{\text{min}}$ then |
| Step 9. remove $q^{K+1}$ //remove subgraphs less than the frequency threshold |
| Step 10. end if                              |
| Step 11. end for                             |
| Step 12. $K = K + 1$                         |
| Step 13. end while//higher order subgraphs are generated in turn until no new subgraphs can be formed |
| Step 14. redundancy screening//screening of redundancy |
| Step 15. return $T/$gets frequent subatlas   |
| Step 16. end                                  |
can set the corresponding minimum frequency threshold according to the personalized requirements of products, and then obtain the common structure that fits the design intention.

4. Conclusions

In this paper, a method for mining common structures of 3D assembly models is proposed. Firstly, the relevant information of the 3D model is extracted and represented by the attribute make-up feature adjacency graph. Then, the relationship between components in the 3D assembly model was analysed, and the isolated attribute assembly feature adjacency graph was combined to form the attribute adjacency graph of the 3D assembly model. Finally, frequent subgraph mining algorithm was used to discover the common structure of the 3D assembly model. The method can from 3D assembly model of product structure, the generality of the excavation support reuse to provide reference for related design work, improve the efficiency of product design and manufacture accuracy as well as design, increase the flexibility of the 3D assembly model information reuse, effectively improve product design and manufacturing enterprise of the reuse of exploring information resources.

4.1. Future Work

(1) In the process of graphical representation of model information, this paper mainly highlights the assembly features of the model and unifies the
nonassembly features in the semantic node representation. However, certain nonassembly features also contain important information about the 3D assembly model. Therefore, how to screen the useful structural features in the 3D assembly model and create a suitable information representation is one of the focuses of the subsequent research.

(2) In the design and manufacturing process of a product, there are many uncertainties in the many objective and subjective needs of designers for 3D assembly model information at different stages and under different conditions. Therefore, subsequent specific analysis of design intent can be carried out to summarise its composition characteristics and construct a more comprehensive and prominent semantic information expression standard for design personality, and this paper mainly focuses on similarity evaluation in design reuse, and the evaluation scope will be expanded subsequently.

Data Availability

The data used to support the finding 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 project was supported by National Natural Science Foundation of China (Grant no. 51705392), Shaanxi Science and Technology Resources Open Sharing Platform (Grant no. 2021PT-006), Shaanxi Youth Science and Technology Rising Star Project (Grant no. 2020KJXX-071), and Shaanxi Provincial Department of Education Special Scientific Research Project (Grant no. 18JS043).

References

[1] S. Chakrabarty and J. Wolter, "A structure-oriented approach to assembly sequence planning," IEEE Transactions on Robotics and Automation, vol. 13, no. 1, pp. 14–29, 1997.
[2] Z. Y. Liu, J. R. Tian, and S. Y. Zhang, "Research on product level information representation for virtual assembly," Journal of Computer Aided Design and Graphics, vol. 13, no. 3, pp. 223–228, 2001.
[3] X. Y. Mou, Method for Assembly Sequence Planning Based on Hierarchical Model and Human-Mputer Collaborative, pp. 99–103, New Technology New Process, Beijing, China, 2015.
[4] Z. Li, Research on Hierarchical Representation for Similarity Assessment and Reuse of 3D CAD Models, Shanghai Jiao Tong University, Shanghai, China, 2015.
[5] G. Y. Yang, L. Tian, D. P. Zhao, and Y. Xin, "Two-dimensional description method of process information for 3D assembly instruction publishing," Applied Mechanics and Materials, vol. 602–605, pp. 3774–3777, 2014.
[6] A. Bourjault, Contribution to a Methodological Approach of Automated Assembly: Automatic Generation of Assembly Sequence, University de Franche-Comté, Besançon, 1984.
[7] L. S. Homem de Mello and A. C. Sanderson, "A correct and complete algorithm for the generation of mechanical assembly sequences," IEEE Transactions on Robotics and Automation, vol. 7, no. 2, pp. 228–240, 1991.
[8] H. Xiao, Y. Li, J. F. Yu, and Z. Jie, "CAD mesh model simplification with assembly features preservation," Science China, vol. 57, no. 3, pp. 1–11, 2014.
[9] M. Zuo, A Method for Common Design Structure Discovery in 3D Assembly Models and Their Design Reuse, Northwestern Polytechnical University, Xi’an, Shaanxi, China, 2016.
[10] S. F. Wu, Design Reuse Method Research for Auto Parts, Zhejiang University, Hangzhou, Zhejiang, China, 2009.
[11] T. Q. Ma, Research on CAD Model Reuse Technologies Supporting Product Rapid Design, Dalian University of Technology, Dalian, Liaoning, China, 2009.
[12] Y. Zhou and J. R. Zheng, "Local matching of assemblies based on subgraph isomorphism and case matching," Journal of Computer-Aided Design & Computer Graphics, vol. 22, no. 22, pp. 299–305, 2010.
[13] Z. Han, R. Mo, Z. Chang, L. Hao, and W. Niu, "Key assembly structure identification in complex mechanical assembly based on multi-source information," Assembly Automation, vol. 30, no. 2, pp. 208–218, 2017.
[14] P. Wang, J. Zhang, Y. Li, and J. Yu, "Reuse-oriented common structure discovery in assembly models," Journal of Mechanical Science and Technology, vol. 31, no. 1, pp. 297–307, 2017.
[15] P. Liu, Y. Li, and K. F. Zhang, "Layered orientation graph modeling method for the assembly of complex products," Journal of Machine Design, vol. 24, no. 4, pp. 33–35, 2007.
[16] J. H. Liu and Y. Z. Hou, "MBD model parametric method based on hybrid-attributed adjacency graph," Journal of Computer-Aided Design & Computer Graphics, vol. 30, no. 07, pp. 155–160, 2018.
[17] A. Inokuchi, T. Washio, and H. Motoda, "An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data," in European Conference on Principles of Data Mining and Knowledge Discovery, D. A. Zighed, J. Komorowski, and J. Zątkow, Eds., pp. 13–23, Springer, Berlin, Heidelberg, 2000.
[18] J. F. Guo, W. Zhang, and R. Chai, New Algorithm of Mining Frequent Subgraph, Computer Engineering, vol. 30, no. 27, pp. 27–32, 2011.
[19] L. Yan, Study on Steady Frequent Subgraph Mining Algorithm, Liaoning University, Shenyang China, 2018.
[20] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large database," in Proceedings of the ACM SIGMOD International Conference on Management of Data, vol. 22, pp. 207–216, ACM Press, Washington, D.C., USA, June 1993.
[21] R. Agrawal and R. Srikant, "Mining sequential patterns," in Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14, IEEE, Taipei, Taiwan, March 1995.
[22] Y. Z. Bao, The Improvement of Apriori Algorithm on the Basis of Rough-Set Theory, Shanghai Normal University, 2010.
[23] A. Inokuchi, T. Washio, and H. Motoda, "Complete mining of frequent patterns from graphs: mining graph data," Machine Learning, vol. 50, no. 3, pp. 321–354, 2003.