A Sub-assembly Identification Algorithm for Assembly Sequence Planning

Yan Cao[1][a], Xiaoxi Kou[1] and Sen Cao[2]

[1]School of Mechatronic Engineering, Xi'an Technological University, Xi'an, Shaanxi, China

[2]Shandong Vocational College of Economics and Business, Weifang, Shandong, China

[a]jantonyz@163.com

Abstract. In order to simplify assembly sequences, reduce assembly difficulty and costs, and guide product assembling process, a sub-assembly identification algorithm is presented and applied to assembly sequence planning problems in the field of intelligent planning. It is used to identify sub-assemblies in the assemblies of different sizes. Through establishing a weighted undirected connected graph, the relations between the parts in an assembly is represented and base parts are determined. The algorithm is designed and realized in Matlab. Its feasibility is verified by a motorcycle assembly instance. It is proved that the sub-assembly identification algorithm can be used to optimize assembly sequences and shorten assembly sequence planning time.

Introduction

The assemblies of different sizes are confronted with combination explosion, too large search space, exponential growth of restraint combinations and so on. In intelligent sequence planning, there also exist difficulties, such as, large number of components, complex structures, high assembling demands and sub-assembly identification difficulty. Through recognizing sub-assemblies of different sizes in a complex assembly, a hierarchical assembling structure of the assembly can be formed. Thus, the complex assembly sequence planning problem is decomposed into several sub-problems, reducing assembly difficulty, facilitating the generation of assembly sequences, and improving assembly planning efficiency [1]. That can better guide product assembling process and reduce assembly costs.

There are many domestic and foreign researches on sub-assembly identification methods. Currently, the identification methods are divided into two categories, i.e., automatic extraction and, human-machine interactive extraction [2]. The automatic extraction methods usually require the establishment of a part correlation model for an assembly [3]. In the paper, a weighted undirected connected graph is established to recognize sub-assemblies in an assembly. A sub-assembly identification algorithm is designed and realized in Matlab.

Weighted Undirected Connected Graph

An assembly generally consists of n parts. $P = \{P_1, P_2, \cdots P_n\}$ is the part set of the assembly. According to the connection relations between the parts, a weighted undirected connected graph G is established as below.

$$G = (V, E, W).\tag{1}$$

Hereinto, $V = \{P_1, P_2, \cdots P_n\}$ includes the parts in the assembly.

$E = \{e_{ij} | (P_i \in V)(P_j \in V), \ i < j\}$ means there exists fastening relationship or contact relationship between the parts.

$W = \{w_{ij} = (r, x_0, y_0, z_0, x_1, y_1, z_1)\}$ represents whether e_{ij} is a fastening edge or a contact edge. If it is a contact edge, its contact direction is also marked.

Weight w_{ij} is defined as follows.

$$
\begin{cases}
\text{If and only if there is the contact relation between } P_i \text{ and } P_j, r=1. \\
\text{If and only if there is the fastening relation between } P_i \text{ and } P_j, r=2. \\
\text{If and only if } r=1 \text{ and } P_i \text{ contacts } P_j \text{ in X direction, } x_0=1. \text{ Otherwise, } x_0=0. \\
\text{If and only if } r=1 \text{ and } P_j \text{ contacts } P_i \text{ in X direction, } x_1=1. \text{ Otherwise, } x_1=0. \\
\text{If and only if } r=1 \text{ and } P_i \text{ contacts } P_j \text{ in X direction, } y_0=1. \text{ Otherwise, } y_0=0. \\
\text{If and only if } r=1 \text{ and } P_j \text{ contacts } P_i \text{ in X direction, } y_1=1. \text{ Otherwise, } y_1=0. \\
\text{If and only if } r=1 \text{ and } P_i \text{ contacts } P_j \text{ in X direction, } z_0=1. \text{ Otherwise, } z_0=0. \\
\text{If and only if } r=1 \text{ and } P_j \text{ contacts } P_i \text{ in X direction, } z_1=1. \text{ Otherwise, } z_1=0.
\end{cases}
\tag{2}
$$

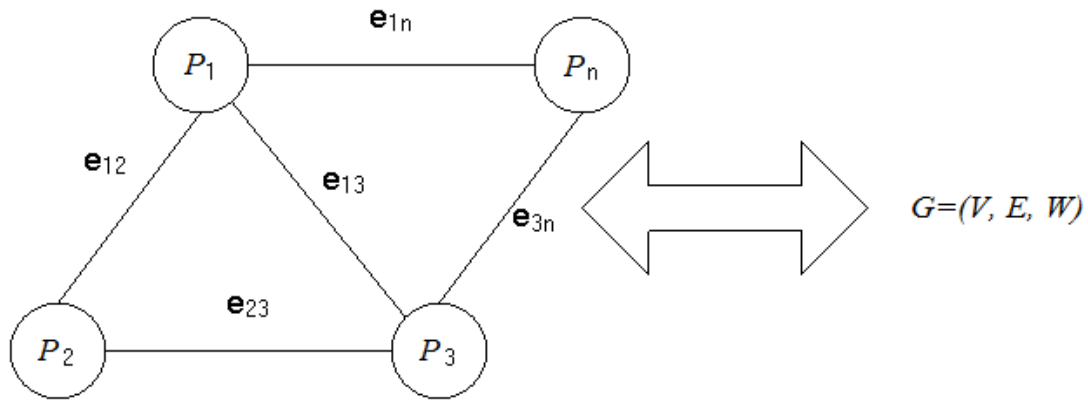The weighted undirected connected graph is shown in Fig. 1.

Figure .1 The weighted undirected connected graph

The weighted undirected connected graph is represented with matrix $C = \left[c_{ij}\right]_{n \times n}$. c_{ij} is a structure variable[4], whose member variables include r, x_0, x_1, y_0, y_1, z_0 and z_1. The value of the element in the matrix C is equal to the weight of the corresponding edge in the graph G [5]. Because i<j in E, matrix C is an upper triangular matrix. Because the contact relation is constructed in an orthogonal coordinate system, by definition of each component of the edge weight, variable x_1, y_1 and z_1 of matrix element c_{ij} can be used to represent the contact relation between parts P_i and P_j in -X, -Y and -Z direction.

Sub-assembly Identification Algorithm

A sub-assembly is a group of parts of stability and independence. The number of the parts in the sub-assembly should be greater than or equal to 2 and less than the total number of the parts in the assembly. The mutual relation between parts is steady and there exists at least one assembly or disassembly direction between them. First, in order to identify the sub-assemblies, construct a weighted undirected connected graph of the assembly. Then, according to the weighted undirected connected graph and the definition of sub-assembly, recognize and divide sub-assemblies. And then, continue to recognize and divide sub-assemblies and adjust the equilibrium between the sub-assemblies [6]. Finally, according to the principle of equilibrium, handle isolated parts and assign them to corresponding sub-assemblies.

In the assembling process, usually a part is chosen to be assembled first that is called the base part. All the other parts are assembled based on the base part. Therefore, any sub-assembly does not include the base part.

The procedure of the sub-assembly identification algorithm is as follows.

(1) Find the minimal loop for each fastening edge in the weighted undirected connected graph of an assembly. Choose the minimal loop whose vertexes do not include base parts to generate the minimal loop set.

(2) Combine the minimal loop sets that have a common vertex until all the loop sets do not have a common vertex. The combined loop sets are called the maximal loop sets.

(3) Judge the global interference between each maximal loop set and all the parts that are not in the maximal loop set. If there is not a such interference part, it is a steady sub-assembly that cannot spontaneously separate, i.e., a base sub-assembly.

(4) If the global interference part of the maximal loop set supports a part in the maximal loop set, add the global interference part in the weighted undirected connected graph into the maximal loop set. Then, go to step (3).

(5) Choose the direction in X, Y and Z that has not been judged. According to the values of the components of every element in the weighted undirected connected graph, generate all contact part sets that do not include the base part in this direction.

(6) If a part in the contact part set belongs to a base sub-assembly, add all the parts of the base sub-assembly into this contact part set.

(7) In the direction that has not been judged, judge according to matrix *C*. If there is a part that does not belong to the contact part set and is not a base part, but it contacts arbitrary two parts of the contact part set in the direction that has not been judged, add the part into the contact part set.

(8) Combine the expanded contact part sets obtained in step (7) that have common vertex until the intersection between the contact sets is empty. The combined sets are called expanded contact part sets.

(9) In a direction beyond the aforementioned directions that have not been judged, judge whether there exists the contact relation between the part that does not belong to the contact part set and the part that belongs to the contact part set. If there are such parts in each direction beyond the aforementioned directions that have not been judged, the expanded contact part set is the sub-assembly that can only be assembled or disassembled as a whole in the assembly, i.e., an expanded sub-assembly. Otherwise, the expanded contact part set is not an expanded sub-assembly in the direction beyond the directions that have not been judged. Assign the current direction as judged.

(10) Judge whether X, Y and Z direction are judged. If so, end the algorithm. Otherwise, return step (5).

Algorithm Development in Matlab

The sub-assembly identification algorithm is designed and developed in Matlab, which identifies sub-assemblies in terms of an assembly.

The application of the algorithm is as below.

(1) Input the part data of an assembly and the number of parts.

(2) Initialize the base parts of the assembly.

(3) Assign parts to the base parts with the strongest correlation. Form part sets.

(4) Judge whether the algorithm converges. If so, output the sub-assemblies. Otherwise, go to (3).

In the paper, a motorcycle assembly example is used to verify the developed algorithm. The identification results of the sub-assemblies in Matlab are shown in Fig. 2. The points in Fig. 2 represent the parts in the assembly. The parts in different regions represent different sub-assemblies.

Conclusions

In the paper, a sub-assembly identification algorithm for assembly sequence planning is presented and studied. A weighted undirected connected graph is used to represent the relations between the parts in an assembly. The sub-assembly identification algorithm is used to identify the sub-assemblies in the assembly. The algorithm is developed in Matlab. A motorcycle example is used to verify the

algorithm. The algorithm can be used to optimize assembly sequences and shorten assembly sequence planning time.
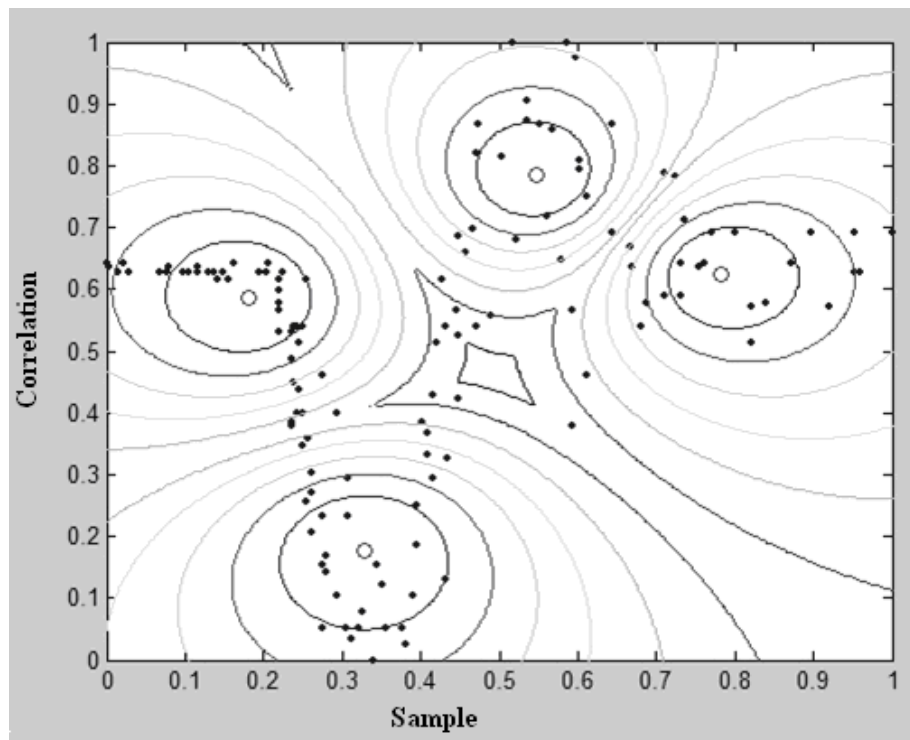

Figure 2 The motorcycle's subassembly recognition results

References

[1] C. Sinanoglu, H.R. Börklü: Assembly Automation Vol. 25 No. 1 (2005), p. 38

[2] Q. Su: Robotics and Computer-Integrated Manufacturing Vol. 25 No. 1 (2009), p. 224

[3] A.J.D. Lambert: European Journal of Operational Research Vol. 168 No. 9 (2006), p. 932

[4] J. Chen, Y. Liu, S. Lu, B. O'Sullivan and I. Razgon: Proceedings of the STOC 2008 Victoria (BC) Canada (2008), p. 177

[5] J. Guo, J. Gramm, R. Niedermeier and S. Wernicke: Journal of Computer and System Sciences Vol. 72 No. 8 (2006), p. 1386

[6] W.C. Chen, Y.Y. Hsu, L.F. Hsieh and P.H. Tai: Expert Systems with Applications Vol. 37 No. 1 (2010), p. 716