A New Graph Representation of Assembly Planning

Dai Guohong
School of Mechanical Engineering
NanJing University of Science & Technology
NanJing, China, 210094

Zhang Youliang
School of Mechanical Engineering
NanJing University of Science & Technology
NanJing, China, 210094

You Fei
School of Mechanical Engineering
JiangSu Teachers University of Technology
Changzhou, China,213001

Abstract - To express nonlinear assembly sequences, a logical tree model for assembly sequences planning is proposed after analyzing the disadvantage of the existing method to generate and express of assembly sequence. Meanwhile, based on the information of the function structure tree and its parts and components of the production, the broad sense sub-assembly is identified by dividing the assembly relation graph and matching the rule. So, the identification of broad sense sub-assembly is simplified. Then, the preferential constraint relation of parts of broad sense sub-assembly is judged by matching assembly knowledge. Finally, based on the function structure tree, the data node and the relation node are continually added. From the top layer to the bottom layer, the tree for expressing assembly sequences is created in turn.

Key words: Assembly Sequence; Tree Model; Broad Sense Sub-assembly; Assembly Knowledge

1.0 Introduction

Assembly sequence planning is the basic of CAAPP (Computer Aided Assembly Process Planning), and is one of main studies on DPA (Digital Pre-Assembly). In concurrent engineering, the designer must evaluate the assembling feasibility of different design projects for one production. So, the assembly sequence of every design project is generated in advance. To study the assembly sequence planning, Homen de Mello and De FaZio proposed Green Model based on the theory of AI (artificial intelligence). Its base principle is that a network map is formed to describe the contact of parts based on the product assembly graph, nodes represent parts, and edges represent relations of parts. Then, some questions on the inter-constraint of parts are asked by the system. Preferential relations of assembly constraint are determined according to user's answer. Meanwhile, the assembly sequence of the product is generated [1-4]. There are some scholars who obtain assembly sequence by using the cut-set method of the graph theory, or genetic algorithms, or simulated annealing algorithm. The problem of assembly sequence planning is solved in a certain extent. But, when a product is made up of numerous parts, large numbers of assembly sequences will be generated because of using the cut-set algorithm, or simulated annealing algorithm. The calculation time is very long, and the searching extent is very large because of using genetic algorithms. So there is very difficult to plan assembly sequence using these methods [5-9].
To express the assembly sequence, Homen de Mello and Sanderson proposed the AND/OR graph model [10]. The optimized object continually changes when the time and the manufacturing environment change because of the complexity and diversification of assembly sequence planning. So the assembly sequence planning is a very complex and nonlinear design. In the product assembly, self-assembly operations of some components are concurrent. To describe these states, an amount of data nodes are included in the AND/OR graph, and they are iterative and tedious. In assembly sequence planning, the searching algorithm is very complex, and the calculation is very large, because the AND/OR graph is a sort of netlike model. So it is not fit to express the nonlinear assembly sequence. In addition, the existing methods for planning assembly sequence mainly base on the mathematic algorithm. As far as the geometrical feasibility is concerned, large numbers of assembly sequence can be generated which don't contain the assembly process information. In the evaluation and optimization of assembly sequence, the calculation is very large, and the illation is very complex.
For these problems, a method is proposed that the assembly sequence is planed layer upon layer according to the function structure tree of the production. Relevant parts of the production are divided to form broad sense sub-assemblies (including sub-assembly and clustering), and the relevant assembly sequence is determined according to assembly process rules. Sequentially, a new tree model of the assembly sequence is created to express these geometry-feasible and

process-reasonable assembly sequences. The generation, evaluation and optimization of assembly sequences are simplified very much.

2.0 Concept of the tree model of assembly sequence

According to the top-down design idea, the function model of the product is created, and the function structural tree is proposed at the beginning of the design. This tree is a hierarchical model, which shows the relation of all parts and components of the product. The product is consisted of parts and components, and the component is consisted of parts and sub-components. There is a sort of the hierarchical relation among the product, components, and parts. This hierarchical relation of parts and components is showed in figure 1. This hierarchical relation contains the information of assembly sequence. It says that the part in the lower layer of the hierarchical model is firstly assembled, and then the upper part is assembled. Secondly, there are some preferential assembly relations among parts and components on the same layer. These relations reflect their assembly sequence [11], and a section of the relation information is contained in the assembly relation graph.

So, a kind of SAP-tree (Sequence, And, Parallel represent three relation nodes) is proposed to express nonlinear assembly sequences based on the function structure tree and the assembly relation graph of the product. The assembly sequence SAP-tree can describe not only the hierarchical relation of parts and components on the different layer, but also the preferential relation of parts and components on the same layer. There are two sorts of nodes on the SAP-tree.

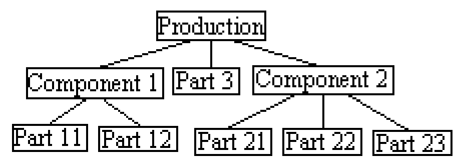


Figure 1 Structural Tree of Product Function

(1) Data Node (d): It is assembling element (AE) or non-assembling element (NAE). AE can be a broad sense sub-assembly, or a part, which is one cell of assembly. NAE is a virtual data node formed with some broad sense sub-assemblies and parts in order to express the SAP-tree structure easily. It is not a broad sense sub-assembly.

(2) Relation Node (r): It expresses the assembly sequence of some data nodes. It includes S-node, A-node, and AP-node.

On the SAP-tree, algorithms of three nodes are described as follows:

1) Sequence Algorithm

In assembly sequence, all data nodes under S-node are arrayed in turn from left to right. In the figure 2(a), the assembly sequence is combined with AE1, AE2, and AE3 in series. There is only one assembly sequence.

2) And Algorithm

In assembly sequence, all data nodes under A-node are arrayed in anyone sequence. In the figure 2(b), the assembly sequence is randomly arrayed with AE1, AE2, and AE3. There are six assembly sequences.

3) AP Mix Algorithm

In assembly sequence, all data nodes under AP-node are arrayed in anyone sequence, and the self-assembly operation of every broad sense sub-assembly, which is represented as the data node under AP-node on the SAP-tree, can be run at the same time. In the figure 2(c), the assembly sequence is randomly arrayed with AE1, AE2, and AE3. There are six parallel assembly sequences, and the self-assembly operation of AE1, AE2, and AE3 can be run at the same time.

According to the description of the node attribute, obviously, the SAP-tree has the better capability to express nonlinear assembly sequence. The definition of assembly sequence SAP-tree is described as follows:

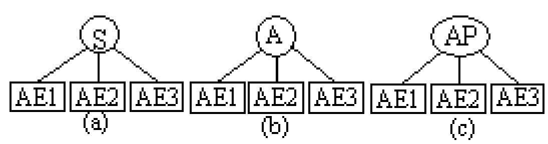


Figure 2 Relation Node of SAP-tree

Assembly sequence SAP-tree is consisted of one origin node, m relation nodes, and n data nodes (m，n≥1). Type( ) is a algorithm for getting the node data. It must meet two rules as follows:

(1) For one relation node, if r∈R, then type(r)∈G, with G ={ S, A,AP };

(2) For one data node, if d∈D, then type(d)∈H, with H={AE, NAE}[12].

On the basis of previous analysis, the SAP-tree is feasibly model for expressing the nonlinear assembly sequence, if AE or NAE is described as a data node. For example, the assembly sequence SAP-tree describes all geometry-feasible and process-reasonable assembly sequences of one assembly in figure 3. On the SAP-tree, the data node 23|45 is NAE, and other data nodes are AEs. Under the AP-node, the sequence of assembling broad sense sub-assembly 23 and 45 into part 1 is random, and self-assembly operations of two data nodes are run simultaneously. One is to assemble part 4 into part 5, the other is to assemble part 3 and part 2 together.

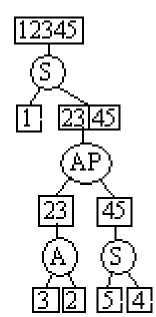


Figure 3 Assembly Sequence SAP-tree

3.0 Method to Create the Tree Model of Assembly Sequence

In the creation of the SAP-tree of assembly sequence, the function structure tree of the product is continually searched layer upon layer. Broad sense sub-assemblies are identified, and they are added onto the SAP-tree as data nodes. Then, the sequence of broad sense sub-assemblies and parts on the same layer are determined by matching assembly knowledge, and the relation node is added too. Finally, a SAP-tree of assembly sequence is created after repeating the upper course continually.

3.1 Broad Sense Sub-assembly Identification

In the assembly sequence generation, the previous method is mainly mathematical operation to deal the product consisted of very finite parts. If the part number of the production is very large, the problem of combinational explosion will appear. In most existing research, the calculation for generating assembly sequence is reduced by dividing sub-assemblies from parts of the assembly[5][13]. If the clustering is divided from parts of the assembly again, the efficiency for generating assembly sequence is more heightened. So, to simplify the assembly sequence generation, the identification of the broad sense sub-assembly is firstly made when the relation netlike graph of assembly is divided according to the function structure tree. After a complex assembly is divided into several broad sense sub-assemblies, the number of planed broad sense sub-assemblies and parts is very small every time. The complexity of creating the SAP-tree is consumedly reduced, and the efficiency of assembly sequence generation is heightened.

3.1.1 Broad Sense Sub-assembly Definition

The broad sense sub-assembly is a part-group to show the relatively concentrative assembly, which is consisted of some parts possessing stable or unstable assembly relation. The broad sense sub-assembly is classified two sorts as follows: One is the sub-assembly. Parts in a sub-assembly have stable contact relation. They won't spontaneously separate. The other is the clustering. Parts in a clustering don't have relatively stable contact relation. They will separate without the constraint action of the outer force. Sub-assembly and clustering are defined as follows:

Definition 1 Let $A = \{P_1, P_2, \cdots\cdots, P_n\}$, $A_1 = \{P_i, P_{i+1}, \cdots\cdots, P_m\}$, with $A_1 \subset A$, $1 \leq i \leq m-1$, $2 \leq m \leq n-1$, and A_1 must meet three rules as follows:

1) The base-part of assembly A is not included in A_1.

2) The connection relation of parts in A_1 is stable (screw connecting, tight fit, etc).

3) On one disassembly direction, all parts, which block A_1 to be disassembled, don't be blocked by A_1. That is to say the assembly operation of parts in A_1 is finished in series.

Then, A_1 is called as a sub-assembly.

Definition 2 Let $A = \{P_1, P_2, \cdots\cdots, P_n\}$, $A_2 = \{P_i, P_{i+1}, \cdots\cdots, P_m\}$, with $A_2 \subset A$, $1 \leq i \leq m-1$, $2 \leq m \leq n-1$, and A_2 must meet three rules as follows:

1) The base-part of assembly A is not included in A_2.

2) The connection relation of parts in A_2 is unstable.

3) On one disassembly direction, all parts, which block A_2 to be disassembled, don't be blocked by A_2. That is to say the assembly operation of parts in A_2 is finished in series.

Then, A_2 is called as a clustering.

3.1.2 Algorithm to Identify Broad Sense Sub-assembly

In the existing researching method, the sub-assembly is identified by the complicated matrix operation based on the relation graph of an assembly. Because the matrix operation aims to all parts of the production, there is enormous data to process yet. If the assembly relation graph is firstly divided based on the function structure tree of the production, then the broad sense sub-assembly is judged, aiming to all parts of the component. Consequently, the algorithm to identify the broad sense sub-assembly is simplified very much. So, to create the assembly sequence SAP-tree easily, the function component must be redivided to generate the broad sense sub-assembly, according to the function structure tree and the relation graph of the assembly. The algorithm is designed as follows:

Step 1 Take out the first layer information of the function structure tree of the assembly.

Step 2 Identify the base-part of the production according to the function component information of the lower layer and the assembly relation graph of the production. Divide the assembly relation graph to generate the base-part and sub-relation graphs or parts.

Step 3 Take out the lower layer information of the function structure tree of the assembly.

Step 4 Judge the attribute of every function component of the current layer in turn.
Step 5 If all sub-nodes of every function component of the current layer are parts, then go to step8.
Step 6 Identify the base-part of this function component according to the information of the function sub-component on the lower layer, then divide the base-part from the relation graph of this function component, and generate some sub-relation graphs or parts.
Step 7 Take out the lower layer information of the structure sub-tree of the function component. Judge the attribute of every function component of the current layer in turn, and go to step5.
Step 8 Identify the base-part of the function component, then divide the base-part from the relevant sub-relation graph, and generate some open-loop relation graphs or parts.
Step 9 If the connection relation of relevant parts of the open-loop is stable, then define them as a sub-assembly.
Step 10 If the connection relation of relevant parts of the open-loop is unstable, then define them as a clustering.
Step 11 If all function components have been judged, then go to step 12, else go to step 4.
Step 12 Output the information of broad sense sub-assemblies, and the algorithm ends.

3.2 Assembly Process Knowledge

In the generation of assembly sequence SAP-tree, preferential constrain relations of broad sense sub-assemblies or parts on the same layer are judged by matching assembly process knowledge. Assembly process knowledge is mainly described as some rules to judge the assembly sequence of all parts of one sub-assembly, or all broad sense sub-assemblies or parts on the same layer. The representation of assembly knowledge is the form of "If …, then …"production rules, and it includes the experience knowledge and the fact knowledge.

3.2.1 Experience Knowledge

The experience knowledge derives from the assembly experience of longtime assembly operation. It is mainly fit to judge assembly sequence of all parts in a clustering. It includes assembly accuracy knowledge, assembly stability knowledge, assembly facility knowledge, and assembly part attribute knowledge.

3.2.2 Fact Knowledge

The fact knowledge derives from the assembly process fact of the product. It is mainly fit to judge assembly sequence of all parts of a sub-assembly, or all broad sense sub-assemblies and the base-part on the same layer. The relative local relations of all parts of a sub-assembly are fastened by the directed-connector. So the assembly sequence of all parts of a sub-assembly is commonly constant. In the assembly sequence, the base-part is always before all broad sense sub-assemblies on the same layer. So the assembly sequence of the base-part and all broad sense sub-assemblies is constant too. The fact knowledge includes assembly constraint knowledge of sub-assembly and judging knowledge of base-part.

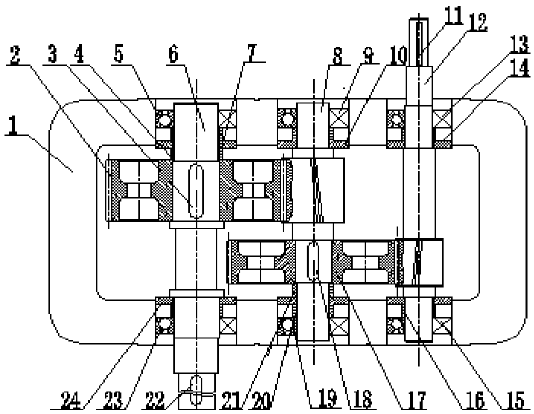
3.3 Algorithm Analysis to Create SAP-tree

In the creation of assembly process SAP-tree, the main operation includes the identification of the broad sense sub-assembly, the information mapping of the assembly relation graph, parts and components, and the algorithm to add the relation node and the data node. The algorithm to create assembly sequence SAP-tree is described as follows:
Input: the function structure tree of the production, the assembly relation graph, the information of parts and components
Output: the assembly sequence SAP-tree
Step 1 Take out the function structure tree, the assembly relation graph, the information of parts and components, then identify the broad sense sub-assembly. Its information is saved.
Step 2 Take out the function structure tree of the production. Create the origin node of the assembly sequence SAP-tree.
Step 3 Take out the node information on the next layer of the function structure tree of the production, and the information of broad sense sub-assemblies and parts are mapped. Add all data nodes of the current layer into the SAP-tree.
Step 4 If one part of this layer is the base-part, then the base-part is separated from the data node containing it, and generate two new data nodes. Add a data node as the father node of all data nodes except the base-part on this layer, and add the relation node S of the base-part and this father data node.
Step 5 Judge the parallelism of self-assembly operations of every data node on this layer.
Step 6 If the self-assembly of all data nodes on this layer is parallel to do, then the relation node AP is added as a father node of all data nodes on this layer, or else the relation node A is added as a father node of all data nodes on this layer.
Step 7 Judge every data node attribute of the function structure tree of the production on this layer in turn.
Step 8 If one sub-node of all data nodes of this layer on the function structure of the production isn't the part, then go to step3.

Step 9 Add every sub-node of all data nodes of this layer on the function structure of the production into the SAP-tree.
Step 10 If one part of this layer on the SAP-tree is the base-part, then add a data node as the father node of all data nodes except the base-part. Add a relation node S as the father node of the base-part and this father data node.
Step 11 Add a data node as the father node of all parts composing every broad sense sub-assembly according to the information of every broad sense sub-assembly contained in all data nodes of this layer, and judge the attribute of every data father node.
Step 12 If one father data node is a sub-assembly, then add a relation node S under this father data node, and rearrange their locations of all nodes under relation node S according to their assembly sequence.
Step 13 If one data father node is a clustering, then add a relation node A under this father data node
Step 14 Add a relation node A as the father node of all father data nodes and part data nodes of this layer.
Step 15 Output the assembly sequence SAP-tree, and the algorithm ends.

4.0 Example

The assembly structure graph of the main body of two-grade speed-ratio cylinder gear reducer is shown in figure 4. There



1-base box 2、17-big gear 3、11、18、22-key 4、21- tube-shaped part
5、9、13、15、19、23-bearing 6-low-speed gear-axle 8-gear-axle1
7、10、14、16、20、24-oil slinger 12-high-speed gear-axle

Figure 4 Structural Graph of Two-grade Speed-ratio Cylinder Gear Reducer

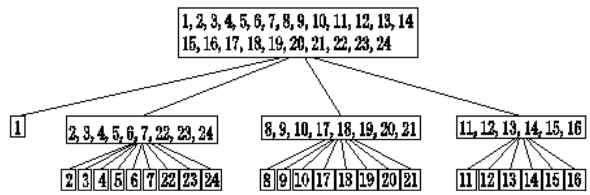


Figure 5 Function Structure Tree of Reducer

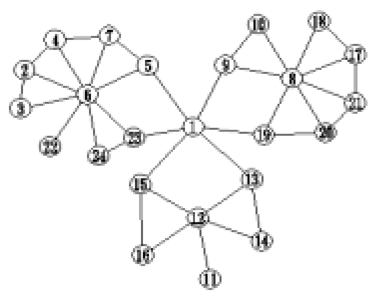
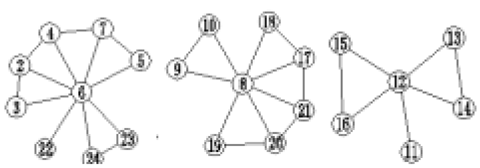


Figure 6 Assembly Relation Graph of Reducer



(a) (b) (c)

Figure 7 Divided Sub-relation Graph

are all twenty-four parts. The function structure tree of the reducer is shown in figure 5, and the assembly relation graph of the reducer is shown in figure 6. According to the function component information of the function structure tree, the assembly relation graph is divided to identify the base-part and the broad sense sub-assembly by matching the assembly process knowledge. Firstly, the base-part 1 and three sub-relation graphs are generated (Figure 7). Then, the base-part of the broad sense sub-assembly is identified for all parts of every sub-relation graph. Base-parts of three sub-relation graphs are respectively part 6, part 8, and part 12. After the base-part is divided from the sub-relation graph, the broad sense sub-assembly is identified for the left link. Sub-assemblies include

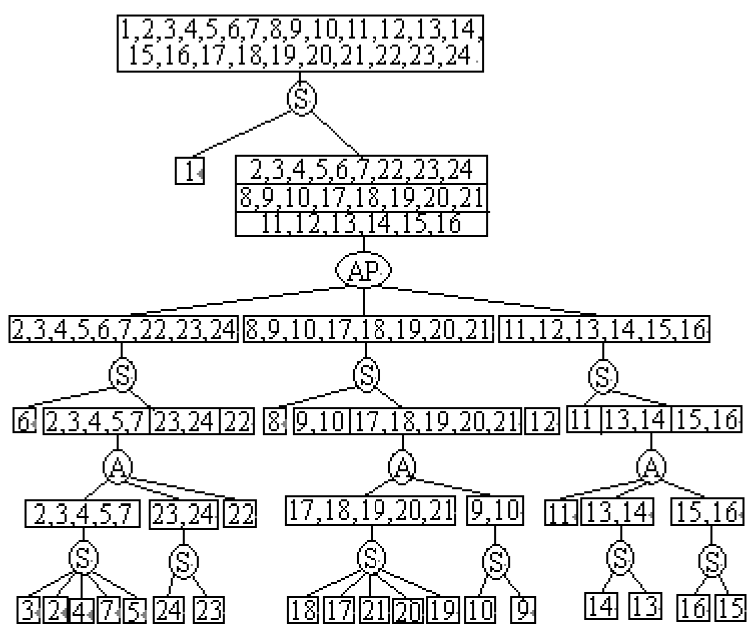


Figure 8 Assembly Sequence SAP-tree of Reducer Main Body

the link 2-3 and the link 17-18. Clustering include the link 4-7-5, the link 23-24, the link 9-10, the link 19-20-21, the link 13-14 and the link 15-16. Two left parts are part 11 and part 22. Lastly, the assembly sequence SAP-tree is planned layer upon layer, according to the function structure tree of the production, and the data node and the relation node are continually added to create the SAP-tree (Figure 8).

5.0 Conclusion

In this paper, the assembly sequence SAP-tree is proposed to compare with the AND/OR graph. The expression of assembly sequence is much more succinct on the assembly sequence SAP-tree. The description ability of the assembly sequence is much stronger, because the SAP-tree can express the parallel assembly sequence. Meanwhile, in the generation of assembly sequence, the searching algorithm of the SAP-tree is very easy and fast, because it is a tree, but the AND/OR graph is a net. In the creation of the assembly sequence SAP-tree, the contained information of the function structure tree of the product is used. The identification of the broad sense sub-assembly is consumedly simplified, because the geometric reasoning method of dividing the assembly relation graph and the knowledge reasoning method of matching the rule are used. Then, based on the function structure tree, the SAP-tree is planned from the top layer to the bottom layer, and the assembly sequence SAP-tree is created by using the method of the knowledge reasoning. The generation efficiency of the assembly sequence is heightened. Because the assembly sequence expressed in the SAP-tree contains lots of assembly process information. These information is the better basic of evaluating and optimizing assembly sequence.

References

[1] Peilin YANG, Jun ZHU, Xiaonan CHEN. "Identification of Sub-assemblies in Sequence Planning". Journal of Xi'an Jiaotong University. 33 (12): 40-43, 1999.

[2] Röhrdanz F, Mosemann H, Wahl F M. "Constraint Evaluation for Assembly Sequence Planning". Proceedings of 1997 IEEE International Symposium on Assembly and Task Planning, Marina del Rey, California. 263-268, 1997.

[3] Gottipolu R B, Ghosh K. "A simplified and efficient representation for evaluation and selection of assembly sequences". Computers in Industry, 50(3): 251-264, Mar 2003.

[4] Lazzerini B, Marcelloni F. "A genetic algorithm for generation optimal assembly plans". Artificial Intelligence in Engineering. 14(4): 319-329, Apr 2000.

[5] Yu-jun CAO, Jianzhong SHANG, Li TANG, Zhixiong ZHANG. "Research on Sub-assembly Identification in Assembly Sequence Planning". Machinery. 31(11): 47-49, Nov 2004.

[6]Bruce.Romney, Cyprien.Godard, et al. "An Efficient System for Geometric Assembly Sequence Generation and Evaluation". In: Proceedings of the 1995 ASME International Computer in Engineering Conference. ASME Press, 699-712, 1995.

[7]Tiam-Hock Eng, Zhi-kui ling, Walter Olson, Chuck Mclean. "Feature-based assembly modeling and sequence generation". Computer & Industrial Engineering. 36: 17-33, 1999.

[8] Bonneville F, Perrard C, Henrioud J M. "A Genetic Algorithm to Generate and Evaluate Assembly Plans". IEEE Symposium on Emerging Technology and Factory Automation.2: 231-239, 1995.

[9] Dini G, Failli F, Lazzerini B, et al. "Generation of Optimized Assembly Sequences Using Genetic Algorithms". Annals of the CIRP, International Institution for Production Engineering Research. 48(1): 17-20, Jan 1999.

[10] Homem de Mello L S, Sanderson A C. "AND/OR graph representation of assembly plans". IEEE Journal of Robotics and Automation. 6(2): 188-199, Feb 1990.

[11] Daqun ZHU, Liangcai ZHAO, Jun ZHANG. "A Method of Generating Assembly Plan Based on Level Hierarchy Connection Relation Model". Journal of East China Shipbuilding Institute.14(1): 71-75, Jan 2000.

[12] Zhijun LI, Guoding CHEN, Wu ZHAO. "Computer Aided Process Planning—CAPP". Chengdu University of Science & Technology Publishing Company, 1996.

[13] Jingxia ZHANG, Runxiao WANG. "Assembly Sequence Generation Based on Sub-assembly Identification". Machinery Design & Manufacture. 36(6): 88-89, Jun 2004.