Chapter 109
Assembly Sequence Planning Based on Assembly Knowledge Database

Rong Li and Ying Tian

Abstract An assembly sequence planning method based on assembly knowledge is put forward. Three types of the assembly knowledge were defined, and the knowledge database was established, which include products' part designing knowledge, joining knowledge, and assembly rule knowledge. According to all of the knowledge, assembly model was building and simplified. Supported by above knowledge assembly model, by firing the state matrix, all the feasible sequence can be obtained. Then, the sequences of the joining were acquired from the knowledge database, which were insert into the feasible sequence, and the assembly sequence were obtained. Finally, a simple case is provided to illustrate the effectiveness of the proposed method.

Keywords Assembly knowledge • Assembly sequence planning • Petri net

109.1 Introduction

With increasing global competition, customer requirements are rising high. To remain in market under such competition, manufacturers must strive continually to reduce assembly time and cost. The determination of proper assembly sequence is critical, because it affects server aspects of the assembly process as well as the finished assembly. Assembly sequence planning (ASP) can help to reduce overall manufacturing assembly time and cost by helping manufacturers reduce the number of

R. Li (✉)
School of Automobile and Traffic Engineering, Heilongjiang Institute of Technology, Harbin 150050, China
e-mail: mizzle0929@126.com

R. Li
State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150050, China

Y. Tian
School of Mechanical Engineering, Tianjin University, Tianjin 300072, China
e-mail: tianying09@163.com

Z. Zhong (ed.), *Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012*, Lecture Notes in Electrical Engineering 217,
DOI: 10.1007/978-1-4471-4850-0_109,

fixture and tool changes or the number of reorientations during the assembly process, and so on. In addition, designer can use assembly planning to verify that a design can be assembled safely without any interference or collision between components.

Recent years, ASP has attracted many researching efforts. Gottipolu and Ghosh [1] generated feasible sequences by analyzing contact and mobility constraints. Laperriere and ElMaraghy [2] generated assembly sequences using geometric and accessibility constraints. Attempts have also been made to generate sequences directly from the CAD model of the assembly. Reddy B. has developed a new approach for generation, representation, and selection of assembly sequence. In the paper, the contact and the translational functions are built, which are extracted from CAD model of the assembly [3]. Smith developed and automated assembly planner using a genetic algorithm to find assembly sequence plans for a product [4, 5]. Mok introduce a method for generating assembly sequences and part mating operations directly from CAD STEP files [6]. These algorithms can be used conveniently on the computer, and all the feasible sequences can be determined. However, as the number of components in a product increases, the number of feasible sequences also increases. For solving ASP, assembly knowledge is considering; then, the methods of based on knowledge Petri net is proposed.

109.2 Constitution of Assembly Knowledge Library

An assembly is a collection of manufactured parts, brought together by assembly operations to perform one or more of several primary functions. ASP begins with the creation of a CAD model to the assembly. But, the model only expresses the geometric models of the individual component that represents the geometry and positions in the world coordinate space. The CAD model provides enough information for graphic display of the assembly, but it is inadequate for assembly planning. Saaksvuori reports that up to 70 % of a designers time can be saved if the existing knowledge base of an organization can be reused for new design [7]. So, the assembly knowledge library is a very important need. According to research of assembly's knowledge, three kinds of knowledge database have been set up. The first is product's part lib, which are gotten together by designers. The second is connecter-structure case base, which react strongly to product assembly. And, the third is rules knowledge base. The assembly knowledge library is the foundation of assembly model and assembly planning in future.

109.2.1 Product CAD Database

Industries now realize that the best way to reduce life cycle costs is to reuse design knowledge. Such reuse of existing designs is beneficial from many different perspectives. It reduces design time by eliminating the need for modeling and analysis for the assembly being reused. Furthermore, the existing CAD model is already tested and has been used in some product successfully. This further reduces the product development time and cost. So, it is necessary to establish parts library with endorse design knowledge sharing. Entity relation of product's part database is set up by IDEF1.

In the parts library, two kinds of parts are constituted; one is the standard parts, and the other is the custom parts. All of the solid models are built based on UG, and the family table library of standard parts is established, the database library of parts is established by using SQL Server 2000, and the application program is developed with VC++6.0 and UG/OPEN, so as to parts library system is established.

109.2.2 Joining Knowledge Database

Joints on a structure are inevitable because of various engineering requirements and products are very rarely monolithic. Existing designer systems have limitations on capturing the nongeometric aspects of designer intent on an assembly with joints. Therefore, the development of an assembly formalism to specify joining relationships symbolically is a prerequisite for an intelligent assembly modeling system.

A relational joining knowledge model that can explicitly describe various logical and physical relationships among the components of the joining is defined as follows:

$$\Sigma = J_m - A_s - T - P_n \tag{109.1}$$

The relational model of an joining knowledge is a three tuples, where $J_m = \{j_1, j_2, \ldots, j_m\}$ is a set of symbols, and each symbol corresponds to joints(standard part) in the joining.

$A_s = \{a_1, a_2, \ldots, a_s\}$ is a set of symbols, and each symbol corresponds to additional part in the joining.

T is the symbol the tool to operate the joint.

$P_n = \{p_1, p_2, \ldots, p_n\}$ is a set of symbols, and each symbol corresponds to custom part in the joining.

The example of joining is shown in Fig. 109.1

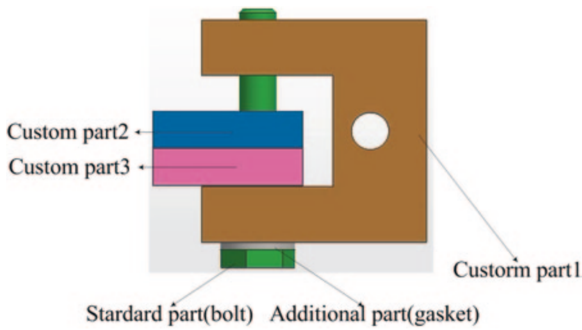

Fig. 109.1 An example of a joining

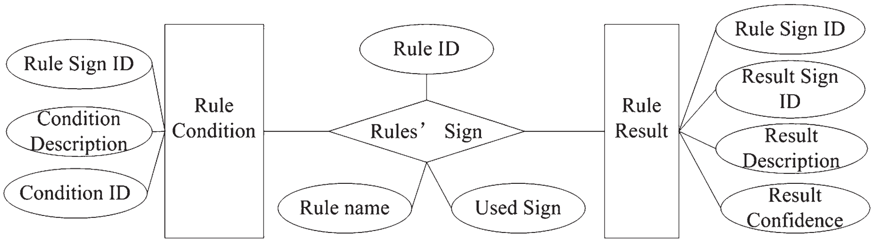

Fig. 109.2 E-R Model of assembly rule lib

109.2.3 Assembly Rule Knowledge

According to the combination of research and related literature, most of the assembly rule knowledge is causal relationship or logical relationship. For example, ODN ≤ 16, $L \geq 25$, IDN > 10 (ODN is the outer diameter, IDN is the inner diameter, and L is the length of the pipe), these rules are referenced by production rule. So, the rules database is established by the object-oriented method (the E-R model of the assembly rule is shown in Fig. 109.2).

109.3 Assembly Model Based Assembly Knowledge

109.3.1 Knowledge-Based Petri Net System

The elementary Petri net is extended to form the knowledge-based Petri net. Knowledge-based petri net (KBPN) is defined by 8-tuples [8]:

$$\text{KBPN} = \left(S, T; F, W, M_0, C_f, K_S, K_T\right) \tag{109.2}$$

where,

$S = \{S_a, S_c\} = (s_1, s_2, \ldots, s_m, s_{c1}, s_{c2}, \ldots, s_{cn})\,(m, n > 0)$ is a finite set of places, and Sa is similar to a set of places in the usual Petri net. Sb is a set of flow-control places, $S_c = (s_{c1}, s_{c2}, \ldots, s_{cn})$. Sc is used to store the part's properties knowledge, which is stored in the products database or the joining database.

$T = \{T_a, T_b\} = (t_1, t_2, \ldots tn, t_{r1}, t_{r2}, \ldots t_{rm})\,(n, m > 0)$ is a finite set of transitions, and Ta is similar to a set of transitions in the usual Petri net. Tb is a set of rules, $T_b = (t_{r1}, t_{r2}, \ldots, t_{rm})$. These rules are stored in the assembly rules database.

$S \cap T = \Phi$, $S \cup T \neq \Phi$; $F \subseteq S \times T \cup T \times S$ is a set of directed arcs, which is used to link places and transitions.

W is weight function on arcs.

$Cf \subseteq (S \times T)$ is arc label of the KBPN, which includes permitted arc labels and inhibited arc labels.

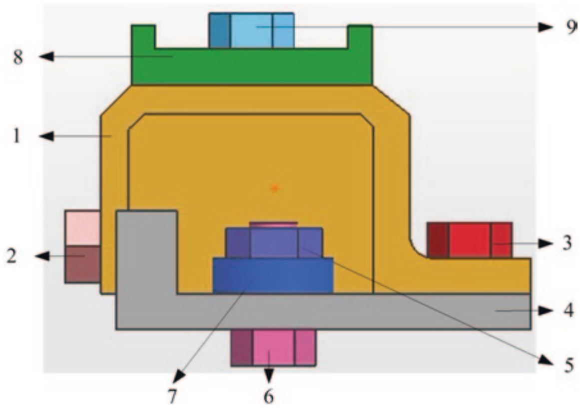

Fig. .109.3 Complicated assembly

K_S represents knowledge associated with places.
K_T represents knowledge associated with transitions. M_0 is initial marking.

109.3.2 Application Example

For the purpose of illustrating the solution procedure developed in the previous section, the assembly product is provided as application example shown in Fig. 109.3. The assembly's KBPN is established in the Fig. 109.4, as we all known.

All of the parts are stored in the product database, and many attributes attach to the part are stored in, too, such as weight, size, class, and so on. They are the design knowledge of the parts. For example, according to the assembly rules knowledge, heavier part is assembled early. Four joining is acquired according to the joining knowledge. So, p_1, p_8, and p_9 consist of one of the joining, which is named as p_{jA}, and p_1 is first assembly in the joining. As the same reason, the p_{jB} represents the p_4, p_5, p_6, and p_7; and the simplified KNPN is obtained in the Fig. 109.5.

109.4 Assembly Sequence Planning

ASP can plan an important role in product design. In this paper, the assembly sequence is obtained by the KBPN. It is assumed that all of the transitions can be fired in the KBPN, so all the feasible assembly sequence may be generated by firing the model. Because of the assembly knowledge, the KBPN is simplified, so the nodes are reduced and the feasible sequences become less. The assembly sequence is obtained by the following algorithm:

1. The T vector is established, which is used to store transition;
2. C is the incidence matrix of the KBPN being established, and the element is indexed sequential data set in the P $\times$ T;

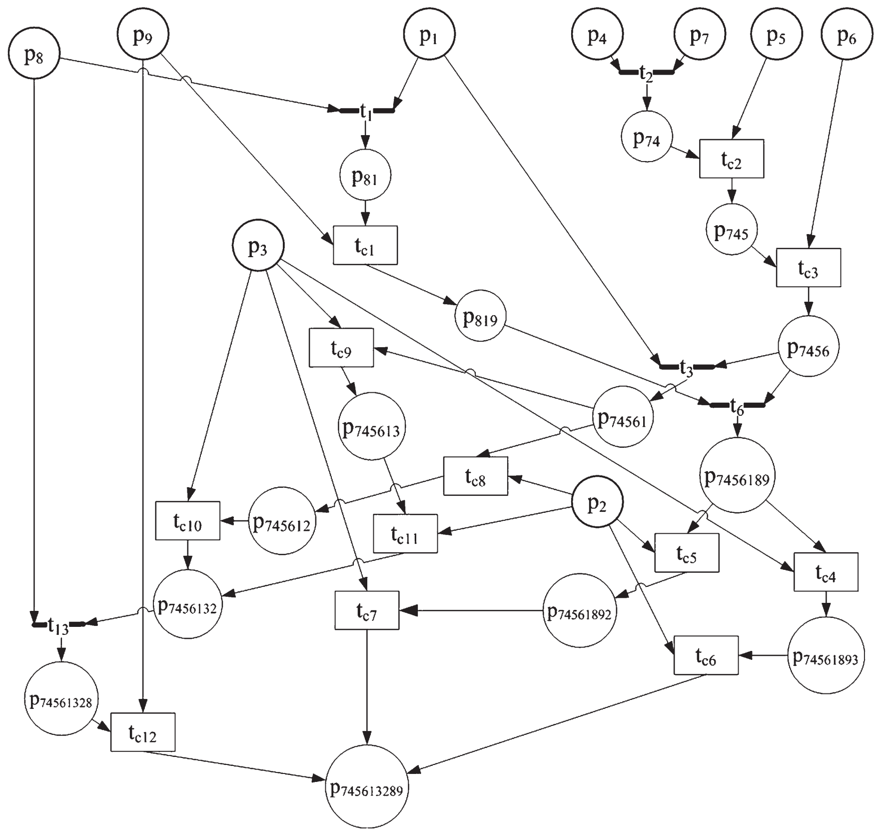

Fig. 109.4 Simplified KBPN of the assembly

Fig. 109.5 Simplified KBPN of the assembly

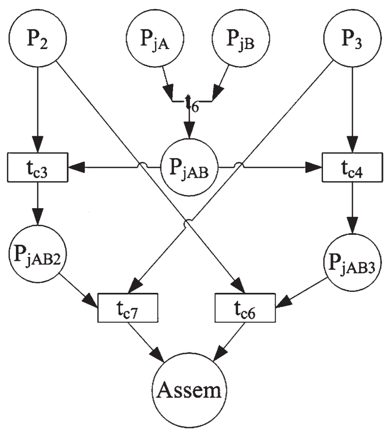

Table 109.1 Relationship Matrix of Fig. 109.5

	t_6	t_{c5}	t_{c4}	t_{c7}	t_{c6}
p_2	0	−1	0	0	−1
P_{jB}	−1	0	0	0	0
P_3	0	0	−1	−1	0
P_{jA}	−1	0	0	0	0
P_{jAB}	1	−1	−1	0	0
P_{jAB2}	0	1	0	−1	0
P_{jAB3}	0	0	1	0	−1
assem	0	0	0	1	1

3. M is the station vector of KBPN, which reference the assembly is finish;
4. The station equation of KBPN is shown in the following equation.

$$M_0 + C \cdot x = M \tag{109.3}$$

where x is the transition vector, $x = \left[x_j\right], x_j \in \{0, 1\}$.

$$c_{ij} = \begin{cases} 1 & \text{if transition } t_j \text{ pointing } p_i \\ -1 & \text{if transition } p_i \text{ pointing } t_j \\ 0 & \text{others} \end{cases}. \tag{109.4}$$

So, the sequence is obtained by solving the equation. For example, the assembly in the Fig. 109.4 and the relationship matrix is shown in the Table 109.1. And, from the Fig. 109.5, we can know that M_0 represents the initial state, and M represents the finial state of the assembly.

$M_0 = [1, 1, 1, 1, 0, 0, 0, 0]\,T$, $M = [0, 0, 0, 0, 0, 0, 0, 1]\,T$. So, the solutions are $x1 = [1, 1, 0, 1, 0]$, $x2 = [1, 0, 1, 0, 1]$

And, they represent the feasible assembly sequence of the assembly. According to the joining knowledge, sequences of the p_{jA} and p_{jB} can be obtained, and then, the final assemblies sequences are obtain by inserting them into the feasible sequence. So, $p_1\, p_8\, p_9 - p_4\, p_7\, p_6\, p_5 - p_2 - p_3$ is one of the assembly sequences.

109.5 Conclusion

This paper discussed a method of using assembly knowledge to plan sequence of assembly. Assembly knowledge is classified into three types, and every one was stored in the different knowledge database. All of the knowledge was used to establish the assembly model, so the KBPN is simplified, and the nodes of the model were reduced rapidly. According to the knowledge, many sequences which break the rules can be removed to improve efficiency of the assembly sequence planning. According to the firing theory, all of the sequences can be obtain by firing the simplified KBPN of the assembly. And then, joining knowledge is acquired from the database, and the sequence of the joining was inserted into the feasible sequences, so the assembly sequences are established. Finally, an example verified that this approach is validity and efficiency.

Acknowledgments Supported by State Key Laboratory of Robotics and System (Harbin Institute of Technology) (SKLRS-2001-MS-08) and national science foundation of china(51105269).

References

1. Gottipolu RB, Ghosh K (1995) An integrated approach to the generation of assembly sequence. Int J Comput Appl Technol 8(3):125–138
2. Laperriere L, ElMaraghy HA (1994) Assembly sequences planning for simultaneous engineering applications. Int J Adv Manuf Technol 21:132–136
3. Gottipolu RB, Ghosh K (2003) a simplified and efficient representation for evaluation and selection of assembly sequence. Comput Ind 15(03):15–20
4. Smith GC, Smith SS-F (2002) An enhanced genetic algorithm for automated assembly planning. Rob Comput Integr Manuf 45(02):29–37
5. Smith GC, Smith SS-F (2003) Automated initial population generation for genetic assembly planning. Int J Computer Integr Manuf 13:39–44
6. Mok SM, Ong K, Wu CH (2001) Automatic generation of assembly instructions using step. Proc IEEE Int Conf Rob Autom 9:32–37
7. Yin ZP, Ding H, Li et al HX (2003) A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. CAD Comput Aided Des 85(01):4–9
8. Li R, Fu YL, Ma YL, Feng HB (2006) Research of the assembly model based on Petri net. Wuhan Ligong Daxue Xuebao 28(S2):393–397