Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2016, 8(10):220-224

Review Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Assembly Sequence Based on Improved Community Structure Algorithm

Cuihong Zhou

School of Information Science and Engineering Hunan City University, China

---

ABSTRACT

This paper proposed a method for assembly sequence planning for modular product. Community structure were used in assembly sequence. Firstly, GN Algorithm was employed to detecting the module of product, Secondly candidate sets of disassembly sequence can be generated according to the module partition evolution process and priority rules, Lastly, the optimization of disassembly sequence could be deduced from the candidate sets of disassembly sequence. To verify the validity and efficiency of the proposed method, a case study on automobile engine is test, and the corresponding result is presented.

Keywords: Disassembly Sequence Planning; Community Structure; Modularization

---

INTRODUCTION

In recent years, product disassembly sequence planning has become one of the most important issues in green design for complex mechanical products. Assembly sequence planning problem is a NP-hard combinatorial optimization problem [1,2]. Generally, with the increase of components in product, the computational complexity of finding optimal disassembly sequence in a large solution space will increase more quickly. Disassembly sequence planning includes generation and optimization, that is to say, by generating and optimization to get a feasible and practical sequence, at the same time meet constraint and cost objectives. Therefore, traditional methods cannot solve this problem effectively. So, heuristic and meta-heuristic algorithms are often used to search optimal solutions at a high efficiency [3].

In order to improve the efficiency of assembly, the structural module hierarchy and design knowledge were used in disassembly sequence planning. In this method, community structure detecting algorithm–GN Algorithm was employed to detecting the module of product, different value of Q corresponding to different result of module partition which will generate a series of module. The module detecting depicts the dynamics of module partition. Using of reverse educing method and knowledge, disassembly sequence can be deduced from the dynamics process of module partition. To verify the validity and efficiency of the proposed method, a case study on automobile engine is test, and the corresponding result is presented.

This paper is organized as follows: In Section 2 we give presents the literature review about assembly sequence; In Section 3 presents the problem description and formulation. After that, Section 4 presents assembly sequence planning based on community structure algorithm and knowledge, the community Structure and detecting algorithm and disassembly sequence generation and optimation are discussed in detail. Finally, sections 5 describe the practical implementation on automobile engine disassembly sequence planning.

PROBLEM FORMULATION AND DEFINITIONS

Zhang Xiufen and Zhang Shuyou [9-10] propose an approach based on bottom-up ideas. Disassembly hybrid graph model is constructed to describe the constraint and disassembly priority relationships among constituting components of product. The problem of disassembly sequence planning was mapped into optimal path-searching problem. Depth first search is carried out to locate the selective component, and then a new object-driven recursive reasoning method is designed to generate possible disassembly sequences which are used to initialize the particles, and particle swarm optimization is used to obtain the optimum disassembly sequence effectively. To solve product disassembly sequence planning problem efficiently, Wang Hui et al [11] transform the disassembly sequence planning into searching optimum path in disassembly feasibility information graph, and proposed an ant colony optimization algorithm to search optimum solutions. A tour of ant represented a possible product disassembly solution. The pheromone of ant was determined by the number of feasible operation nodes of its tour. It was by two steps to get heuristic information: defining heuristic vector -representing the feasibility of solution- and solving heuristic information-representing the quality of solution. Xue Junfang[13] introduce an improved ant colony optimization algorithm to build disassembly hierarchy information graph and search the optimal solutions.

Liu [14] proposes a disassembly sequence planning approach with an advanced immune algorithm incorporated with the partheno genetic algorithm. An advanced support matrix is proposed to conduct stability judgment, and the stability is used as one of the evaluation objectives, thus the disassembly sequence can be evaluated more comprehensively.

To solve the combination explosion during disassembly sequence generation process, the modularization was used in disassembly sequence planning [18]. ZHOU Xi-mei [16] introduce a disassembly model of product based on modularization and defines restriction matrix, studies on the generating of products' disassembly sequence based on restriction matrix and obtains the method of generating product's disassembly sequence. Guo Weixiang17 proposes an approach based on the modularization. In his approach, a modularization disassembly model was constructed based on the hierarchy network graph of product. Yao liying18 presents a new method of generating disassembly sequence planning based on level construction model and provides rules of generating disassembly sequence. The new theory accords with practice, and improves the rate of calculation.

D.Hu et al [19] use the information provided by the mating features of parts in the product to find the candidate parts for disassembly and to carry out disassembly path planning. A complete and accurate interference checking approach is used to ensure no global collision while disassembling apart. In some cases, it cannot be implemented by geometric reasoning alone, so a set of criteria and heuristic rules based on knowledge, constraints, relationships among parts, and quantitative disassemblability assessment are used. It can also be carried out interactively by the user when necessary. The proposed method is integrated with the CAD model of the product. The user can visually disassemble the product while planning, so it is easier to carry out the disassembly planning and generate an optimal sequence.

Shana S. Smith and Wei-Hsiang Chen [20] present a rule-based recursive method for finding a near-optimal heuristic selective disassembly sequence for green design. The proposed method establishes certain heuristic disassembly rules to eliminate uncommon or unrealistic solutions. In addition, the developed method only considers geometric relationships between apart and its neighboring parts. As a result, the developed method can effectively find a near-optimal heuristic solution while greatly reducing computational time and space.

For a complex maniacal product with a set of components, sequencing the components can be a combinatorial and explosive problem, and it is further complicated to evaluate the efficiency of disassembly sequences, so, using of heuristic algorithm merely is not effective and efficient in solving disassembly sequences planning.

MATHEMATICAL MODE

The general problem of disassembly sequence is described as follows: assumption that a product formed by a set of components $C=\{C_1,C_2,C_3,\cdots,C_n\}$, joined by a set of joist $J=\{J_1,J_2,J_3,\cdots,J_n\}$. The optimum disassembly sequence planning problem consists in determining the order in which these joints are to be broken and the components removed so as to minimize the costs incurred. Thus, as a solution to the problem is a permutation of

the union of the components set and the joints set. However, although a solution is a permutation of C_j, not all such permutations are valid sequences, since, for example, a component cannot be removed before all the joints that affect that component have been released or if the component cannot be accessed. Likewise, a joint cannot be broken before all the components that obstruct access to that joint have been removed.
A solution to the problem is thus a permutation $S = X_1 + X_2 + \cdots + X_{m+n}$ of n m elements in which each position implies the breaking of a joint J or the release of a component C from the rest of the subassembly. We shall represent as $SR(X_k) = X_{k+1}, \cdots, X_{m+n}$. the set of all the elements to the right of X_k in a permutation S. The set of all elements to the right of X_k in a permutation S. We shall represent as J_i the set of joints involving component C_i.

Note that not all the permutations represent a feasible solution to the problem. A solution S will be feasible only if two feasibility conditions are fulfilled. (1) all the joints J_i involving a component C_i must be accessible at the moment of removing that component. Furthermore, all the joints in the J_i must be appearing before C_i. (2) access to the removed component C_i cannot be fully restricted by the rest of the components sill remaining in the subassembly.

Although there is some academic research that considers multi-objects of disassembly sequences, the usually considered goal is to minimize the total disassembly time. In this paper we consider the practical feasibility and the total time of disassembly.

COMMUNITY STRUCTURE AND DETECTING ALGORITHM

A network is a set of nodes and edges, as to a modular product, the nodes represent the components and the arcs establish the relation between the components, and the network of product contracture is created by this way. the edges connect between nodes. It is widely assumed that most social networks show "community structure", in a community structure of network, groups of vertices that have a high density of edges within them, while with a lower density of edges between groups. As to the network of modular product, it is in the same way, components have a strong relation within structure nodules of product, while a weak relation between modules. So, the modules of product have the same way with the community structure of network to some extent, communities in a product component network might correspond to structure modules. Based on this precondition, the community structure detecting algorithm may be employed in module partition of product.

Up to the present, there are many methods for the community structure detecting, and the GN algorithm is the pioneer of divisive algorithms in these method. The GN algorithm, which takes the edge betweenness as the weight of edges, is an effective method for community structure detecting. Practical studies show that the performance of the GN algorithm is quite effective when it is applied to the networks whose community structure can be easily recognized. Thus, in this paper, the GN algorithm was used to the structure modules partition for modular product. By this way, a set of disassembly sequence were generated and provided for planning. The general form of our community structure finding method based on GN algorithm i s as follows21:

(a)Creating the network of product component.
(b)Calculate betweennesss cores for all edges in the network.
(c)Find the edge with the highest score and remove it. If two or more edges tie for highest score, choose one of them at random and remove it.
(d)Recalculate betweenness for all remaining edges.
(e)Repeat from step (c).

The betweenness of an edge is defined as the number of shortest paths between pairs of nodes passing it. The algorithm to calculate edge betweenness is discussed in detail elsewhere2.
In this, betweennesss is normally calculated as the fraction of shortest paths between node pairs that pass through the node of interest, and it is a measure of the centrality of a node in a network. Suppose that gist is the number of

geodesic paths from vertex s to vertex t that pass through i, and suppose that n is the total number of geodesic paths from s to t. Then the betweenness of vertex i is

$$bi = \frac{\sum_{s<t} g_i^{(st)} / n_{st}}{\frac{1}{2} n(n+1)} \quad (1)$$

Where n is the total number of vertices in the network. We may, or may not, according to taste, consider the end-points of a path to fall on that path; the choice makes only the difference of an additive constant in the values for bi22.

Finally, we used modularity Q to measure the result of structure modules partition. The modularity is defined as:

$$Q = \sum_i (e_{ii} + a_i^2) \quad (2)$$

Where e_{ii} denotes the number of edges that fall in the community i, and a_i denotes the number of edges that link to the nodes in community i. If the edges in a network lie no better than randomly, the value of Q is closed to 0. On the other hand, values close to 1, which is the maximum, indicate a strong community structure in networks. For a given network, the higher the value of Q is, the more reasonable the division is in some sense. Generally, the value of Q falls in the range from 0.3 to 0.7. What more, different value of Q corresponding to different result of module partition, which will generate a series of nodule, which can depict the dynamics of module partition. Using of reverse educing method, we can get the disassembly sequence from the dynamics process of module partition.

EXPERIMENT

The candidate result still is series of disassembly sequence, on one hand, of the components located in a module need to be further determined, and on the other hand, their practical feasible still to be checked. (12→16→15→20→14→11→13→5→8→18→4→2→9→10→19→17→1→6→3→7).

Table 1: Seven modules of automobile engine

Module serial	Inclusive components
M1	2,15,16
M2	20
M3	11,13,14
M4	1,3,5,6,7,8
M5	18
M6	4
M7	2,9,10,17,19

Table 2: Candidate Sets of Disassembly Sequence for automobile engine

No.	Candidate Sets of Disassembly Sequence
1	(12,15,16)→20→(11,13,14)→18→4→5→8→(2,9,10,17,19)→(1,3,6,7)
2	(12,15,16)→20→(11,13,14)→18→4→5→8→ (1,3,6,7)→(2,9,10,17,19)
3	(12,15,16)→20→(11,13,14) →8→5→18→4→(2,9,10,17,19)→(1,3,6,7)
4	(12,15,16)→20→(11,13,14) →8→5→18→4→(1,3,6,7)→(2,9,10,17,19)
5	(12,15,16)→20→(11,13,14) →5→8→18→4→(2,9,10,17,19)→(1,3,6,7)
6	(12,15,16)→20→(11,13,14) →5→8→18→4→(1,3,6,7)→(2,9,10,17,19)

So, using of a set of criteria and heuristic rules based on knowledge, constraints, relationships among parts, and quantitative disassemblability assessment, we find the final result of disassembly sequence for the given automobile engine is follows.

CONCLUSIONS

In this paper, only the module level is considered in the disassembly sequence planning. But, the components comprised in connatural module are not considered. In the future study, how to decide the disassembly sequence of components in bottom layer is also an import problem. Additionally, how to further mining value information from the module partition dynamic evolution model provided by community structure detecting algorithm and use those information to aide disassembly sequence planning are also need to be researched.

REFERENCES

[1] ME Kurz; RG Askin. *Int J Production Res*, **2001**, 39(16), 3747-3769.
[2] L Zonglei; X Tao. *IJACT*, **2012**, 4(14), 17-25.
[3] Q Chen; Y Xiang; XX Zhang; X Guo; P Wang. *IJACT*, **2012**, 4(14), 105-114.
[4] G Dobson, *Operation esearch*, **1992**, 43(4), 736-749.
[5] A Allahverdi; J Gupta; T Aldowaisan. *Int J Management Sci*, **1999**, 27(2), 219-39.
[6] M Weng; J Lu; H Ren. Int J Production Economics, **2001**, 70(3), 215-226.
[7] SL Janak; CA Floudas. *Industrial and Engineering Chemistry Res*, **2006,** 45(18), 6190-6209.
[8] A Alle; JM Pinto. *Ind. Eng. Chem. Res.*, **2002**, 41(11), 2689-2704.
[9] A Alle; LG Papageorgiou; JM Pinto, *Computers and Chem Engg*, **2004**, 28(15), pp.3-15, 2004.
[10] PM Franqca; M Gendreau; G Laporte, Int J Production Economics, **1996**, 43(1), 179-89.
[11] DW Kim; KH Kim; W Jang; FF Chen, *Robotics and Computer Integrated Manufacturing*, **2002**, 18, 223-231.
[12] E Sebastian; MG Sequeira; P Luis, *Ind. Eng. Chem. Res.*, **2003**, 42(8), 1761-1772.
[13] MA Shaik; CA Floudas, *Ind. Eng. Chem. Res.*, **2007**, 46, 1764-1779.
[14] F Glover, M Laguna, Tabu search, Kluwer Academic, Boston, **1997**.
[15] P Hansen; B Jaumard, Computing, **1990**, 44, 279-303.
[16] M Nawaz; EE Enscore; I Ham, *Omega*, **1983**, 11, 91-95.