Int J Adv Manuf Technol (2010) 51:1139–1147
DOI 10.1007/s00170-010-2682-7

ORIGINAL ARTICLE

Product cooperative disassembly sequence planning based on branch-and-bound algorithm

Xiu Fen Zhang · Shu You Zhang

Received: 24 November 2008 / Accepted: 15 April 2010 / Published online: 21 May 2010

Abstract This paper focuses on the cooperative disassembly sequence planning (CDSP) problem which is essential to disassembly a large and complex product in practice. A disassembly hybrid graph model (DHGM) is constructed to describe the mating contact and noncontact priority relationships among constituting product components. Thus, the disassemblability constraint expression can be deduced from it by reasoning. The CDSP is different from the disassembly sequence planning except for they are NP-complete. Therefore, a novel method is presented to generate cooperative disassembly hierarchical tree (CDHT) from the DHGM based on branch-and-bound algorithm in which two user-defined variables are introduced to control the size of the CDHT .The optimal solutions with a minimal disassembly time are derived according to the objection function. Finally, an example illustrates the proposed method.

Keywords Cooperative disassembly · Disassembly sequence planning (DSP) · Disassembly hybrid graph model (DHGM)

X. F. Zhang · S. Y. Zhang (✉)
State Key Laboratory of Fluid Power Transmission and Control,
Zhejiang University,
Room No. 408 of the First Teaching Building,
Hangzhou, People's Republic of China 310027
e-mail: xxff_6188@yahoo.com.cn

X. F. Zhang
The College of Mechanical Engineer,
Inner Mongolia University of Technology,
Hohhot, People's Republic of China 010051

1 Introduction

Disassembly is of growing importance with the advent of stringent regulation and environmental consciousness. Disassembly is a process for systematically separating a product into its constituent parts, subassemblies, or other groupings, which is a prerequisite for reuse, recycling, or remanufacturing, since most products should be disassembled in order to be recycled or remanufactured as secondary parts or materials.

Disassembly can be classified into sequential disassembly and parallel disassembly according to the logical sequence to remove parts or subassemblies from a product [1]. In the sequential disassembly, only one part or subassembly is disassembled at a time, while in the parallel disassembly, two or more parts or subassemblies can be disassembled at the same time. Anyway, the task is supposed to be performed by one manipulator. In reality, cooperation between two or more manipulators is important to perform a large and complex product disassembly task in order to reduce the total work time. That means several manipulators can perform different tasks in parallel when they work on the same product, which is so-called cooperative disassembly. Cooperative disassembly has many advantages such as the increase of the flexibility of the system, the increase of the productivity, the increase of the load capacity, and so on.

Disassembly sequence planning (DSP), which tries to find an optimal feasible disassembly task sequence, is studied widely because the sequence of disassembly tasks can drastically affect the efficiency, and the optimal sequence requires fewer changes of tools, fewer changes

of disassembly directions, and less disassembly time than others. In this paper, we focus on the cooperative disassembly sequencing problem, which is important, but dealt with by few papers in the past to the best of our knowledge. The cooperative disassembly sequence planning (CDSP) problem is different from the traditional DSP (shown in Table 1).

Therefore, a novel method is needed to resolve the CDSP problem. First, a disassembly hybrid graph model (DHGM) is constructed from the product 3D CAD model. Then, a method for generation of cooperative disassembly hierarchical tree (CDHT), i.e., generation of (near-)optimum feasible disassembly sequences, based on branch-and-bound algorithm with two user-defined variables, are presented.

The remainder of this paper is structured as follows: Section 2 reviews the related work. Section 3 provides a DHGM as the base of CDSP. Section 4 describes the approach for the generation of CDHT based on branch-and-bound algorithm in detail. Section 5 illustrates the application of the proposed approach. Section 6 ends up with conclusions.

2 Related work

The research area of disassembly sequence planning has been receiving increasing attention in the research literature. In terms of sequential disassembly sequence planning, many different approaches based on metaheuristic and heuristic methods are developed. Among them, graph-based approaches such as component-fastener graph [2], AND/OR graph [3], Disassembly Petri Net [4], cluster graph [5], and so on [6] appear to be popular in the early days. Although these graphs are both complete and well suited for mathematical analysis, their size tends to become unmanageably large when realistic products are considered [7]. Therefore, in recent years, many efficient techniques have been presented for DSP problems, e.g., scatter search [8], genetic algorithm [9], ant colony algorithm [10, 11], algorithm of self-guided ants [12], genetic programming [13], etc.

Other than sequential disassembly, in practical situation, it is often necessary to remove multiple components from a given assembly [1], and an assembly was classified into parallel assembly and sequential assembly by the logical complexity of its disassembly tree (DT) and by the geometric complexity of the nodes in DT [14]. And for parallel assemblies, Dutta and Woo [1] presented a parallel disassembly algorithm based on a disassembly tree. In order to extend the application area of the parallel disassembly algorithm, i.e., both totally and partially order assemblies can be dealt with in parallel, Chen and Oliver et al. [15] presented the Onion Peeling parallel disassembly algorithm based on a mating graph. The main idea is to begin from outside and proceed inwards by merging all internal components together and removing the outside components which can be found according to mating graph. The onion peeling algorithm is employed just for 2D assembly. Kang and Lee et al. [16] presented a method to find a parallel disassembly sequence based on an extended process graph which was obtained by transforming the AND/OR graph. In order to get the optimal solution, an integer programming formulation was developed with the objective of maximizing the overall profit. The challenge for disassembly theory, particularly the extension of its methods to increasingly complex product and more realistic assumptions, has been driven by the need for both environmentally conscious and economically feasible processing of end-of-life products.

Torres et al. [17] presented a product representation based on the hierarchical relations among components that define new components (assemblies), and different disassembly operation rules were obtained from the product representation. A DSP sequence was deduced from the product graph and operation rules. Based on the DSP sequence, Díaz et al. [18] used decision trees to plan the cooperative tasks for a cooperative disassembly robotic system. The parallel disassembly can detach more than one component at the same time only when these components can compose of a subassembly, which lacks of flexibility. This paper deals with a new DSP problem named CDSP, which means a product's disassembly tasks can be accomplished in cooperative way by several manipulators to reduce the total disassembly time.

Table 1 Difference between CDSP and DSP

No.	Difference	DSP	CDSP
1	The number of parts disassembled each time	Certain	Uncertain
2	The number of the manipulator	One	More than one
3	Work areas interference	Not exist	Exist
4	Disassembly task type	Single and static	Multiplex and dynamic
5	The relationship between sequence and basic disassembly time of parts	Sequence time-independent	Sequence time-dependent

The CDSP is a very challenging problem because of the reasons as follows:

1. The length of disassembly sequence is not constant as traditional one.
2. The feasible disassembly sequences are nonlinear.

Branch-and-bound algorithm is one of the main tools in solving NP-hard discrete optimization problems. Gungor A and Gupta SM [19] used a branch-and-bound approach to solve the DSP problem and achieved a near-optimum DSP for a complex product. Benefited from this paper, a modified branch-and-bound algorithm is used to solve the CDSP problem based on DHGM.

3 Disassembly hybrid graph model

3.1 Definition and generation rules

A DHGM aims to describe the product topologized structure. A DHGM can be defined as a 4-tuple: $G = \{V, E_f, E_{fc}, E_c\}$, where node $V = \{v_1, v_2, \cdots, v_n\}$ stands for a minimal disassembled unit (part or subassembly) that cannot be further disassembled; $E_f = \{e_{f1}, e_{f2}, \cdots, e_{fk}\}$ is an undirected edge, which denotes the disassembly contact constraints between two components; and $E_{fc} = \{e_{fc1}, e_{fc2}, \cdots, e_{fcm}\}$ is a directed solid edge, which represents the disassembly precedence and contact constraints between two components; and $E_c = \{e_{c1}, e_{c2}, \cdots, e_{ck}\}$ is a directed dotted edge, which represents the precedence relationship between two noncontact components. An example is shown in Fig. 1 to understand the DHGM. The nut 1 and bolt 4 contact with each other and nut 1 must be removed prior to bolt 4. Therefore, a directed solid edge between nut 1 and bolt 4 is used to represent the relationship, denoted by edge <1,4>. In addition, edge (2,3) or (3,2) represents that spacer 2 and bearing cap 3 just contact with each other without precedence. The directed dotted edge <3, 5> means that bearing cap 3 must be removed prior to bearing 5 though they are noncontact.

Given a 3D model of a product or the 2D drawing, its DHGM can be constructed by a man–machine interactive way. The constraints of constituting components can be taken automatically from the 3D CAD system. The DHGM is further modified according to the following rules: (1) If there is just simple contact without precedence constraints between two components, do nothing, else (2) the edge belongs to E_{fc}. (3) If a component collides with another noncontact component when it is removed, then the relation belongs to E_c. (4) The base part is the last one to disassemble.

3.2 CDSP description

According to the DHGM, the problem of CDSP can be defined as follows:

Given the description of a product, $G = \{V, E_f, E_{fc}, E_c\}$, and the prime disassembly time for each node, $T = \{t_1, t_2, \cdots, t_n\}$, the manipulator set $M = \{m_1, m_2, \cdots, m_k\}$, how to determine a feasible disassembly sequence subject to (1) minimizing the total disassembly time, (2) maximizing parallelism in disassembly, (3) each manipulator can perform only one disassembly task at a time. (4) The different disassembly operations of one part cannot be processed simultaneously.

In a DHGM, a typical disassembly operation is the detachment of a single component, which needs to disconnect all the edges linked to this node.

3.3 Disassemblability constraint condition

Two matrices are deduced from the DHGM, which can be defined as follows:

1. The relation matrix, denoted by M_r.

$$M_r = \begin{bmatrix} mr_{11} & mr_{12} & \cdots & mr_{1n} \\ mr_{21} & mr_{22} & \cdots & mr_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ mr_{n1} & mr_{n2} & \cdots & mr_{nn} \end{bmatrix} = \{mr_{ij}\}_{n\times n}.$$

Where, $mr_{ij} = \begin{cases} 1 & (i,j) \in E_f or \langle i,j\rangle \in E_{fc} or \langle j,i\rangle \in E_{fc} \\ 0 & else \end{cases}$

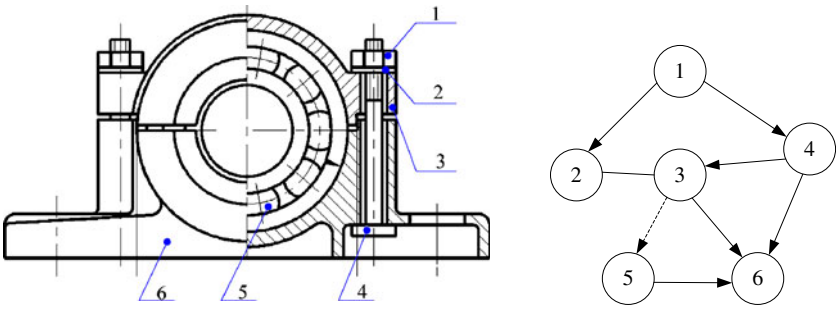


Fig. 1 Roller bearing block (*left*) and its DHGM (*right*)

2. The constraint matrix, denoted by M_{con}.

$$M_{\text{con}} = \begin{bmatrix} mc_{11} & mc_{12} & \cdots & mc_{1n} \\ mc_{21} & mc_{22} & \cdots & mc_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ mc_{n1} & mc_{n2} & \cdots & mc_{nn} \end{bmatrix} = \{mc_{ij}\}_{n\times n}.$$

Where, $mc_{ij} = \begin{cases} -1, & \langle i,j\rangle \in E_c \text{ or } \langle i,j\rangle \in E_{fc} \\ 0, & \text{else} \end{cases}$.

For Roller Bearing Block (Fig. 1), M_r and M_{con} can be deduced from Fig. 1

disassembling →

$$M_r = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

,

disassembling →

$$M_{con} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Disassemblability is to check if a unit can be taken out without colliding with the rest of the product. For node j, mc_{ij}=0 means j can be removed prior to node i. In short, the disassemblability constraint condition of node j can be defined as follows:

$$\sum_{i=1}^{N} mc_{ij} = 0, \quad \sum_{i=1}^{N} mr_{ij} > 0 \tag{1}$$

4 Cooperative disassembly sequence planning

The objective of DSP is to identify the order of disassembly operations (i.e., the order of the nodes in the DHGM) that will minimize the disassembly time and maximize the profit. In order to reduce the complexity of the problem, some assumptions are made as follows:

1. Manipulators' work areas collision is not considered here.
2. All the cooperative tasks begin at the same time.
3. Only nondestructive complete disassembly is considered in this paper.

4.1 CDSP object function

To assess the disassembly sequences, the disassembly time is considered here, which relates to the disassembly cost. The basic disassembly time of every component is supposed to be given, and the additional time such as the time to change tool and direction of movement during disassembly is not considered here. For sequential disassembly, the sum of the basic disassembly time of every part is constant, but it is sequence-dependent for the cooperative disassembly. Let t_i define the basic disassembly time associated with the ith node in the DHGM. Therefore, the cooperative disassembly sequence planning objective function f_u can be defined as follows:

$$f_u = \min \sum_{m} \max_{i\in[1,k]} (t_1, t_2, \cdots, t_i),$$

Where,

m the length of sequence.
k the number of manipulators taking part in the product disassembly task whether they are man or robot [20].

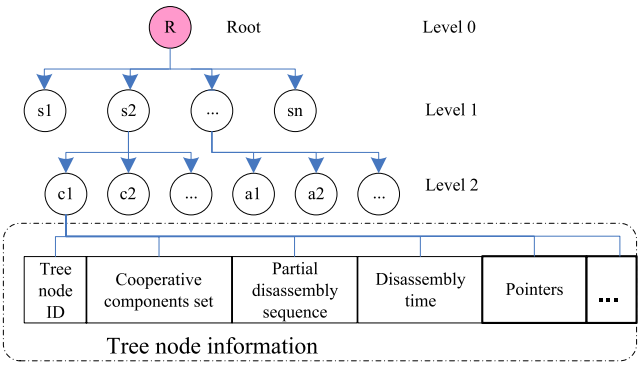


Fig. 2 Illustration of CDHT

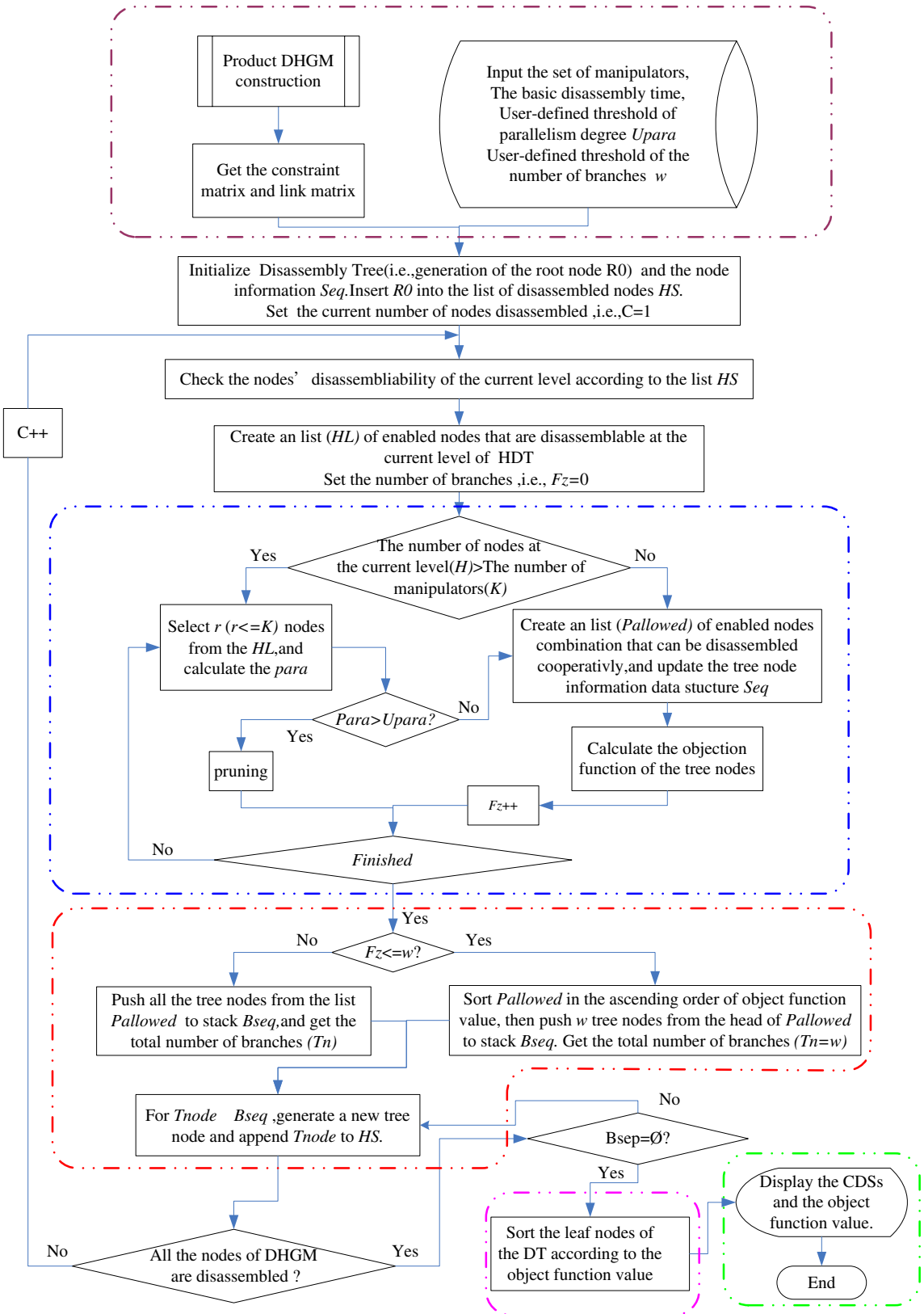


Fig. 3 Flow chart of CDHT algorithm

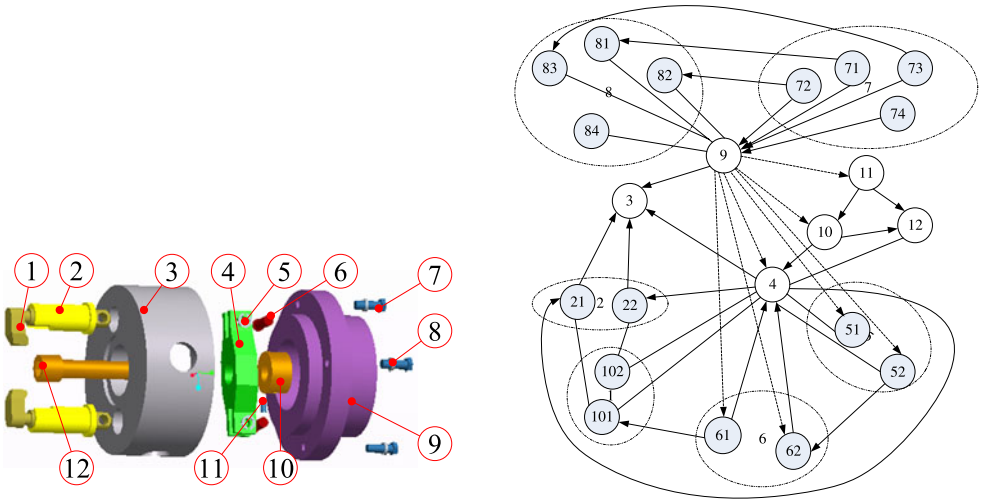


Fig. 4 The explosion view of a valve cover head fixture (*left*) and its DHGM (*right*)

4.2 Generation of (near-)optimal CDSP based on branch-and-bound algorithm

The feasible disassembly sequences are represented as a tree data structure named CDHT in the current paper. A tree is a nonlinear structure like a real inverted tree. The top level node is called *root*, and the rest of the nodes are divided into m ($m \geq 0$) disjoint subsets, each of which is a tree. A node is called the *father* of its lower level subset elements in the tree, and in turn, all the branch nodes of a node is called its *children*. A node without children is a *leaf*.

In the CDHT, each node has a data structure which stores the following information: tree node ID; the feasible cooperative components set, i.e., the disassembly tasks to be performed in a cooperative way at the same time; the list of the tree nodes from the root to the current node, i.e., a partial disassembly sequence; disassembly time of tree node which is the maximum basic disassembly time among the cooperative components set (see Fig. 2).

4.2.1 Branching and bounding constraints

An enumeration of all feasible sequences is a high-combinatorial complex problem; therefore, it is limited in the large problems. Instead of considering all possible sequences, it may be promising to limit the solution space by certain strategy. Based on the DHGM, the branch-and-bound algorithm is used to get the (near-)optimal sequences. Suppose the number of manipulators k is 2; then, not more than two nodes can be removed at one time.

To control the size of the dynamic disassembly tree, one allows two user-defined variables: (1) the parallelism distance of cooperative disassembly tasks, denoted by *para*, where $\text{para} = |\max(t_1, t_2, \cdots, t_i) - \min(t_1, t_2, \cdots, t_i)|_{i \in [1,k]}$.

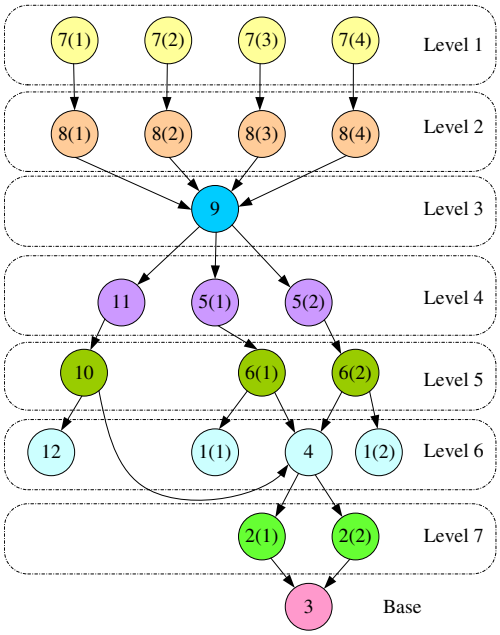
For example, assuming there are four tasks A–D, the basic disassembly time spent for them is as follows: t_a=5 s, t_b=7 s, t_c=2 s. t_d=8 s. Then, the parallelism distance for (A–D) is 6. In addition, the parallelism distance *para* controls the branching procedure when there exist more than two disassembly units, any two nodes cannot be disassembled in a cooperative way if their *para* is larger than the desired parallelism distance; (2) The number of branches at each level, denoted by *w*. With the increasing complexity of the product, the number of branches will become too large to deal with during a feasible computing time. The *w* can be used to define how many near optimal branches are to be chosen each time. The larger the size of *w*, the greater is the chance of finding the optimal solution and the more time-consuming the process.

4.2.2 CDHT algorithm

The overall algorithm for finding a (near-)optimal cooperative sequence is given by CDHT algorithm, which

Table 2 Component list for valve cover head fixture

Part no.	Component name	Disassembly time (unit: seconds)	Quantity
1	Hook-like clamp	1	2
2	Sleeve	1	2
3	Jig body	0	1
4	Hinge plate	2	1
5	Elastic ring	1	2
6	Pin	0.5	2
7	Screw	2	4
8	Washer	0.5	4
9	Packing	5	1
10	Lock nut	2	1
11	Cock screw	3	1
12	Spherical screw	5.5	1



Note:i(j)represents the *jth* component related to graph node *i*,e.g.7(1) represents one of the four screws which is denoted by 7 in the DHGM.

Fig. 5 The disassembly precedence chart

contains three procedures (see Fig. 3), and the steps are described in detail as follows:

Step 1: Construction of DHGM and initialization.

DHGM is the foundation of the CDSP, which can be obtained from the CAD system (e.g., Pro/Engineering) in an interactive way. Input the set of manipulators and the basic disassembly time of each component in the DHGM and initialize the two user-defined variables: parallelism distance *Upara* and the number of branches *w*.

Step 2: Generation of CDHT based on modified branch-and-bound algorithm.

Step 2.1: Initialize the CDHT (i.e., generate the root node, *R0*) and set the root node information *Seq*. Insert *R0* into the list *HS* which contain the removed nodes.

Step 2.2: Disassemblability analysis is performed for the current tree node according to the list *HS*. Create a list (*HL*) of enabled nodes that can be disassembled in the current level of the CDHT.

Step 2.3: Check if the number of nodes in *HL* is larger than the number of manipulators. If it is true, go to step 2.4. Otherwise, go to step 2.5.

Step 2.4: Cluster the components which can be disassembled in a cooperative way under the maximizing parallelism condition (i.e., cooperative parallelism distance $para \leq Upara$).Otherwise, prune it.

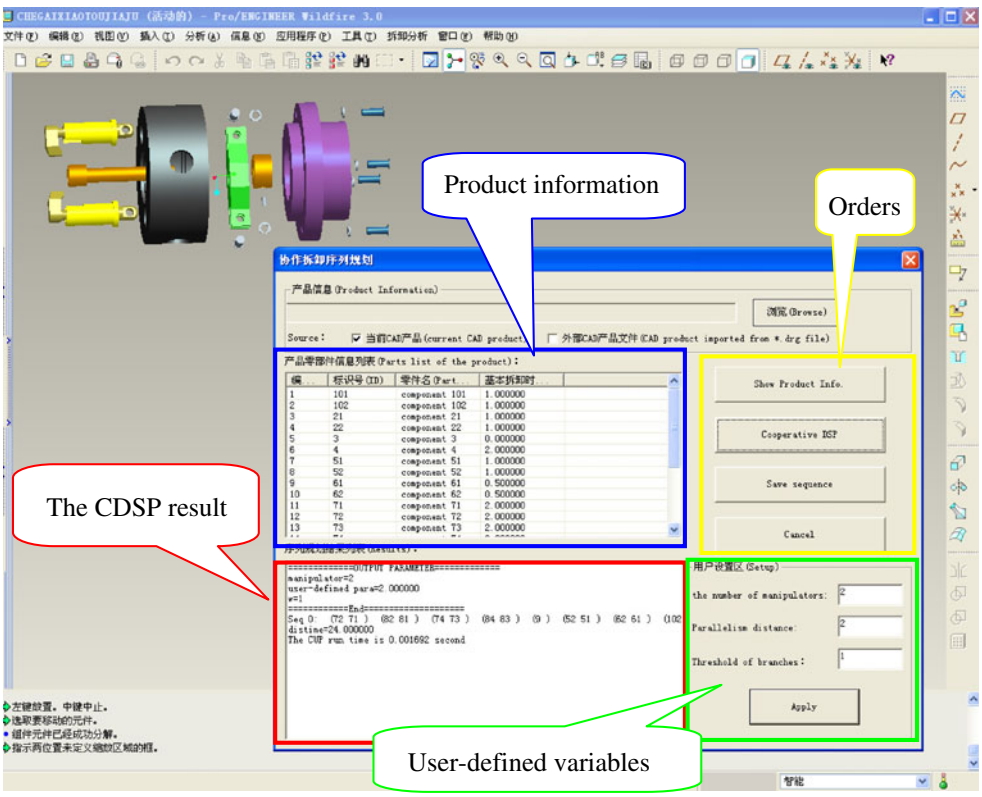


Fig. 6 CDSP results (*para*=2, *w*=1, *k*=2) and system interface

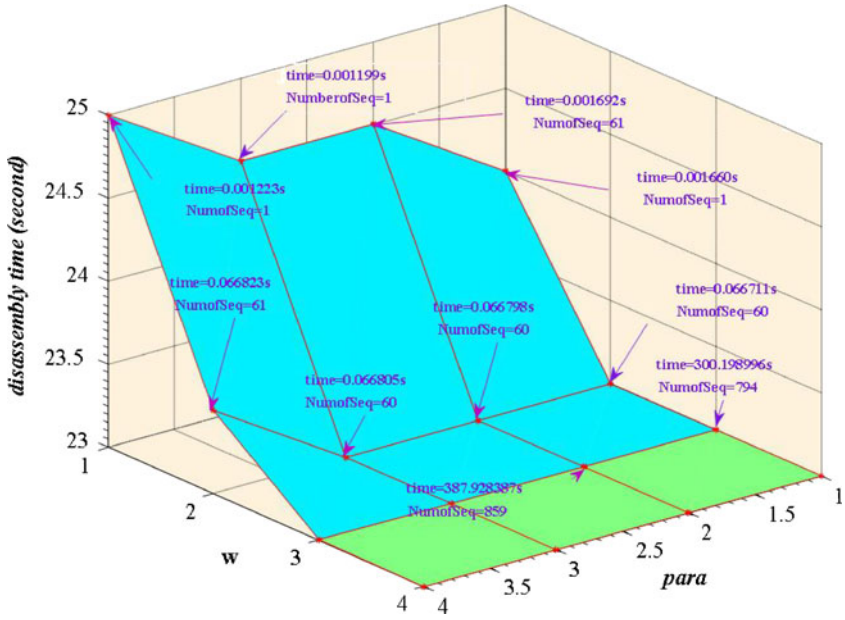


Fig. 7 The relationship between CDSP result and variables (i.e., *para* and *w*)

Step 2.5: Create a list (*Pallowed*) of enabled nodes combination that can be disassembled cooperatively. Insert the node combination into the CDHT as new tree node and update related information data structure *Seq*.

Step 2.6: Calculate the objective function of the partial sequence.

Step 2.7: Check if all the possible clusterings are finished, if it is true, go to step 2.8. Otherwise, go to step 2.4.

Step 2.8: Prune again by the number of branches at each level (i.e., check if the current number of branches (*Fz*) is larger than the threshold (*w*)). If it is true, sort the tree nodes in *Pallowed* according to the objective function value and push the *w* optimum tree nodes into a stack *Bseq*. Otherwise, copy all the feasible tree nodes from *Pallowed* onto a stack *Bseq*.

Step 2.9: Pop a tree node from the *Bseq* (i.e., *Tnode* ∈ *Bseq*), append the *Tnode* to *HS* and go to step 2.2. A branch is constructed until all the components in the DHGM are removed.

Step 2.10: Check if the *Bsep* is empty (i.e., *Bseq* = ϕ); *if it is false, go to step 2.9. Otherwise, the CDHT has been constructed; go to step 2.11.*

Step 2.11: Sort the leaf nodes in the ascending order of their objective function value and copy the best leaf nodes onto the list *BestLeaf*.

Step 3: Display the *BestLeaf* and related information such as the cooperative disassembly sequences, objective function value, and so on.

5 Application

In this study, the proposed method has been experimented on a valve cover head fixture which contains 42 parts. Fig. 4 (left) shows the exploded view .Using the procedure described in Section 3.1, the DHGM representation for the valve cover head fixture is constructed (Fig. 4 right). The component information is listed in Table 2, and the fixture body 3 is the base. It is assumed here that two manipulators take part in the disassembly work. Fig. 5 is a disassembly precedence process chart derived from the DHGM (Fig. 4 right) using the disassemblability analysis formula (1) in Section 3.3.

The CDHT algorithm was implemented using Visual C++6.0 and Pro/toolkit in the Windows XP environment. The results are presented in Fig. 6.

The computational complexity of CDHT depends on the complexity of the product structure and the relationships among the components. The lower bound of the number of the nodes in the CDHT is $n+1$ when there is only one sequential disassembly. However, the upper bound U_{CDHT} of the number of the nodes in the CDHT can be derived when there is no precedence relationship between components. U_{CDHT} is defined as:

$$U_{\mathrm{CDHT}} = \sum_{q=0}^{\lfloor n/m \rfloor} \frac{n!}{(m!)^q (n-qm)!} + \frac{n!}{(m!)^{\lfloor n/m \rfloor}} (n - \lfloor n/m \rfloor m)$$

where $\lfloor n/m \rfloor$ can be achieved when n/m is rounded down to the nearest integer, q is the current level in the CDHT, n is the number of components in the assembly, and m is the

number of manipulators. For example, in this case, $U_{\mathrm{CDHT}} = 1 + C_6^2 + C_6^2 C_{6-2}^2 + C_6^2 C_{6-2}^2 C_{6-4}^2 = 196$ for an assembly with six parts when the number of manipulators are 2. Therefore, the two user-defined variables (i.e., *para* and *w*) are necessary for the real product. Experiments are carried out to analyze the relationship between the result and the variables for the valve cover head fixture (i.e., *para* and *w*).The results are presented in Fig. 7.

From the Fig. 7, the (near-)optimal result is generated with the disassembly time 23 s, and the worse result is 25 s. The computation time will go up when *para* and *w* increase, in which, the result is better. Moreover, the number of branches *w* has a close relationship with the result, and *para* has little effect on the result.

In short, the case shows that the CDHT algorithm can work well for the real product. But it is hard to specify a good selection criterion for *para* and *w*. They can be defined depending on the problem set and the availability of the computational resources by user.

6 Conclusions

CDSP is significant in practice. A systematic approach has been presented in this paper. The main contribution of this paper can be listed as follows:

1. A novel representation of product named DHGM is proposed to describe the constraints relationship between components in the product. From the DHGM, it is easy to deduce the disassemblability formula by reasoning.
2. Based on DHGM, an optimum or (near-)optimum CDSP was generated by utilizing a branch-and-bound heuristics that allows the disassembly time as the basis for optimality. For complex product, two user-defined variables are used aiming to reduce the computational complexity of the heuristics, which is flexible and practical because user has the right to balance the tradeoffs between the quality of the results (e.g., optimum or near optimum) and the computational time.
3. For practical engineering, this cooperative disassembly sequence planning is proposed as a solution for large and complex products.

Acknowledgements This research was supported by the National Key Technology R &D Program, China (No. 2007BA F13B02), the Natural Science Foundation of Zhejiang Province, China (No. Z107416), the National Natural Science Foundation of China (No. 50775201), and the PhD Programs Foundation of Ministry of Education of China (No. 200803350031).

References

1. Dutta D, Woo TC (1995) Algorithm for multiple disassembly and parallel assemblies. ASME J Eng Ind 117:102–109
2. Zhang HC, Kuo TC (1997) A graph-based disassembly sequence planning for EOL product recycling. Twenty-first IEEE/CPMT International Electronics Manufacturing Technology Symposium. Piscataway, New Jersey, pp 140–151
3. Homem de Mello LS, Sanderson AC (1990) AND/OR graph representation of assembly plans. IEEE Trans Rob Autom 6:188–199
4. Moore KE, Gungor A, Gupta SM (2001) Petri net approach to disassembly process planning for products with complex AND/OR precedence relationships. Eur J Oper Res 135(2):428–449
5. Kaebernick H, O'Shea B, Grewal SS (2000) A method for sequencing the disassembly of products. CIRP Ann 49:13–16
6. Li JR, Khoo LP, Tor SB (2002) A novel representation scheme for disassembly sequence planning. Int J Adv Manuf Technol 20:621–630
7. Lambert AJD, Gupta SM (2008) Methods for optimum and near optimum disassembly sequencing. Int J Prod Res 46(11):2845–2865
8. González B, Adenso-Díaz B (2006) A scatter search approach to the optimum disassembly sequence problem. Comput Oper Res 33:1776–1793
9. Kongar E, Gupta SM (2006) Disassembly sequencing using genetic algorithm. Int J Adv Manuf Technol 30:497–506
10. McGovern SM, Gupta SM (2006) Ant colony optimization for disassembly sequencing with multiple objectives. Int J Adv Manuf Technol 30:481–496
11. Wang JF, Liu JH, Li SQ, Zhong YF (2003) intelligent selective disassembly using the ant colony algorithm. Artif Intell Eng Des Anal Manuf 17:325–333
12. Tripathi M, Agrawal S, Pandey MK, Shankar R (2008) Real world disassembly modeling and sequencing problem: optimization by algorithm of self-guided ants (ASGA). Robot Comput-Integr Manuf 25:2–12
13. Li XY, Shao XY, Gao L (2008) Optimization of flexible process planning by genetic programming. Int J Adv Manuf Technol 38:143–153
14. Woo TC, Dutta D (1991) Automatic disassembly and total ordering in three dimensions. ASME J Eng Ind 113:207–213
15. Chen SF, Oliver JH, Chou SY, Chen LL (1997) Parallel disassembly by onion peeling. ASEM J Mech Des 119:267–274
16. Kang JG, Lee DH, Xirouchakis P, Persson JG (2001) Parallel disassembly sequencing with sequence-dependent operation times. CIRP Ann-Manuf Technol 50(1):343–346
17. Torres F, Puente ST, Aracil R (2003) Disassembly planning based on precedence relations among assemblies. Int J Adv Manuf Technol 5(21):317–327
18. Díaz C, Puente S, Torres F (2006) Task planner for a cooperative disassembly robotic system. Inf Control Prob Manuf 2006:161–166
19. Gungor A, Gupta SM (2001) Disassembly sequence plan generation using a branch-and-bound algorithm. Int J Prod Res 39(3):481–509
20. Kim HJ, Kernbaum S, Seliger G (2009) Emulation-based control of a disassembly system for LCD monitors. Int J Adv Manuf Technol 40:383–392