Individual and Group Disassembly Plan Sequence Generation: Theory and Implementation

Young-Q Lee, Soundar R. T. Kumara and Inyong Ham
Department of Industrial and Management Systems Engineering
Intelligent Design and Diagnostics Research Laboratory
The Pennsylvania State University, University Park, PA 16802, U.S.A

요 약

한 제품의 구성 부품 수가 증가함에 따라 조립/분해 순서의 방법 수는 기하급수적으로 증가한다. 또한 이 순서의 선택에 의하여 조립/분해의 효율성이 결정된다. 본 논문에서는 조립제품의 조립순서 (Assembly Sequence) 와 특정 부품의 효율적 교체 (Part Replacement)를 위한 분해순서 (Disassembly Sequence)에 관한 새로운 Algorithm들을 다음 3 가지 경우를 통하여 소개한다 : 1) Full Assembly / Disassembly, 2) Part Replacement by Individual Disassembly, 그리고 3) Part Replacement by Group Disassembly. 이들 Algorithm들은 'Freedom and Interference Space' 라는 새로운 개념으로부터 유도되며, Group Disassembly는 Carpenter's Approach를 통하여 실현 된다. 이 논문에서 소개되는 Algorithm들은 조립/분해 순서와 함께 조립/분해 방향도 제시하는데 비교적 복잡한 예를 통하여 이들을 실현함으로써 실제 제품 Design에의 활용 가능성을 보여준다.

I. INTRODUCTION

In manufacturing, assembly is a significant cost item and is often the most labor intensive. Thus the field of automatic assembly to reduce these substantial assembly cost is one of the largest robot application areas and potentially highly attractive area.There have been several efforts to further reduce the assembly cost and time by designing products in such a way that they can be assembled more easily and by developing systems and machines which make it possible to implement efficient assembly processes. One such effort concentrates on the assembly/disassembly sequence generation; as this domain is essential to the successful implementation of automatic assembly. As the process of assembly/disassembly is reversible, efforts are geared towards disassembly sequence generation. De Mello and Sanderson have introduced AND/OR graph to represent all the possible assembly sequences of a given product. The graph is generated by trying all the possible ways of decomposing all possible subassemblies in a product [3]. Wolter has used Assembly Constraint Graph to generate AND/OR graph [9]. De Fazio and Whitney have introduced a method for generating all possible assembly sequences by obtaining the precedence constraint among liaisons through a question and answer procedure [1]. Several other researchers have described other methods for the representation and generation of disassembly sequences [4, 5, 7, and 10]. For the complex assemblies, however, most of them seem to be very difficult to implement for practical applications.

In this paper, a new approach and its implementation for the generation of disassembly plans which include disassembly directions as well as disassembly sequences are introduced through freedom and interference spaces for the cases of: 1) Full Disassembly / Assembly, 2) Part Replacement by Individual Disassembly, and 3) Part Replacement by Group Disassembly.

II. APPROACH: FREEDOM AND INTERFERENCE

1. Freedom and Interference Space

Disassembling a particular part from an assembly depends on the degree of freedom (FD) of the part and the degree of interference (ITF) which is acting on that part from other parts. In our study, we will assume that our disassembly system has motions of a single translation (a single step) [7]. For the simplicity of explanation we will consider 4-directional disassembly along the X and Z axes ('LURD' in Figure 1 ; L : Left, U : Up, R : Right, and D : Down) in 2-D plane. The directions are subject to available disassembly directions of a system. That is, the number of directions and approach angles are decided by the disassembly situation. Two more directions, backward and forward, could be added in case of 3-D space. If an assembly is on a table then a direction into the table will not be considered. According to the concept of 'Design for Assembly', a system with minimum assembly directions through a single translation is recommended [2]. Therefore, our assumptions are believed not to limit the applications of this study from the practical point of view. Figure 1 shows an example assembly and some relationships of FD and ITF on part y. Subsequently the notations are described.

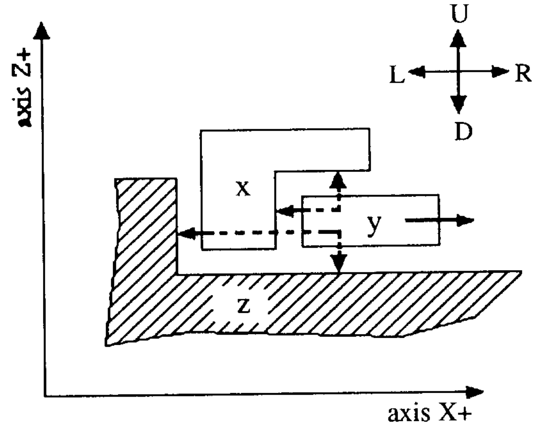


Figure 1. An Example Assembly

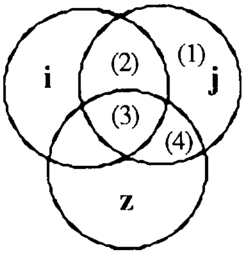
i = a part (or a group of parts) which has ITF on j
j = a subject part (or a group of parts) which we are interested in its degree of FD
A(i,j) = FD of j without ITF of i
A(j,j) = FD of j with all ITF's of the other parts
B(i,j) = FD of j only with ITF of i
C(i,j) = ITF on j from i
D(i,j) = ITF on j only from i (pure ITF on j from i)
ABS = absolute FD (ITF) = 'LURD'
z = a group of all parts in an assembly except x and y (z = z1, z2, . . . , zn).

Part x and y in the Figure 1 correspond to i and j in the notation respectively. Generally, a situation can be represented by the FD and ITF space as shown in Figure 2. From the figure, we can derive the relationships between parts. That is,

$$B(i,j) \cup C(i,j) = ABS$$
$$A(i,j) = B(z,j)$$

$$A(j,j) = A(i,j) - D(i,j)$$
$$D(i,j) = A(i,j) \cap C(i,j)$$

where $B(z,j) = B(z1,j) \cap B(z2,j) \ldots \cap B(zn,j)$



(1) = A(j,j)
(2) = D(i,j)
(4) = D(z,j)
(1) + (2) = A(i,j)
(1) + (4) = B(i,j)
(1) + (2) = B(z,j)
(2) + (3) = C(i,j)
(3) + (4) = C(z,j)

Figure 2. FD and ITF Space

From these we can see that if we are able to identify FD (or ITF) of one part in an assembly against another part, then we can derive A, C (or B), and D. In other words, if we know B, then A, C, and D can be calculated from it directly.

2. Matrices and Sweeping Table

Considering A(i,j) and C(i,j), we can understand them as AND/OR conditions for disassembly. The disassembly of any one of i's of A(i,j)'s prior to the disassembly of j guarantees the disassembly of j. Therefore, A(i,j) indicates 'OR' condition for the disassembly of j. Meanwhile, C(i,j) denotes the directions of ITF of i that should be removed so that j can be disassembled. Therefore, C(i,j) indicates the 'AND' condition for the disassembly of j.

In order to generate disassembly sequences, it is necessary to know the AND/OR relationships between parts. Therefore, A and C values will be derived through B values. We can build up B values for every part of an assembly by using a matrix form. This matrix can be easily achieved because it represents parts relationships of one to one part, not one to many parts. Therefore, we do not need to consider the relationships of a given part(j) with all the other parts. B and C matrices for the example assembly in Figure 1 are given in Figures 3.a and b. Dot ('•') is used to represent 'Nil'.

j i	x	y	z
x		RD	LD
y	LU		LD
z	UR	UR	

a. B Matrix

j i	x	y	z
x		LU	UR
y	RD		UR
z	LD	LD	

b. C Matrix

j Dir.	x	y	z
L	z	xz	•
U	•	x	xy
R	y	•	xy
D	yz	z	•

c. Sweeping Table

Figure 3. B, C matrices and Sweeping Table

From the C matrix we can see that the part y can be disassembled in the direction of 'L' if the disassemblies of parts x and z are completed. Such facts are summarized for each part and for every disassemblable direction and they are used to construct the 'Sweeping Table' as shown in Figure 3.c.

Parts within a cell of the table have the 'AND' relationship (C values) with each other. However, the relationship between cells for a given subject part is 'OR' (A values). This table will be used to generate disassembly sequences.

III. DISASSEMBLY SEQUENCE GENERATION

Once a sweeping table is obtained from the C matrix, sequences for 1) full disassembly (F.D.A), 2) part replacement by individual disassembly (I.D.A) and 3) part replacement by group disassembly (G.D.A) can be generated.

1. Full Disassembly (F.D.A)

F.D.A implies disassemblies of all parts in an assembly one by one. This is known as an 'Onion Peeling' [10]. It is implemented by the following algorithm.

```
algorithm F.D.A
   do-until all parts are disassembled
      - disassemble all parts with ' • ' in the sweeping table
        in available directions.
        ( if such a part does not exist, no sequence for F.D.A exists )
      - arrange a new sweeping table by deleting disassembled parts.
   end-do
end-algorithm
```

2. Part Replacement by I.D.A

In this section and the following section, a disassembly sequence generation for the replacement of a particular part is discussed. It generates disassembling sequences and disassemblable directions in order to disassemble a particular part from an assembly. It could be utilized for the replacement of an out-of-order or worn-out part in repair and maintenance. As there is no need to disassemble an assembly completely in this case, it generates sequences which gains access to the objective part directly. A temporary assumption is that only I.D.A which means disassembling one part at a time is available. The algorithm for part replacement by I.D.A is as follows:

```
algorithm part replacement by I.D.A
   - if the objective part is disassemblable then stop
   - delete redundant cells in the sweeping table
     (for two cells i and j in a column, if cell i is a subset of cell j then
      cell j is redundant)
   - let the objective part be a node of level 0 in a disassembly tree

   do-until  1) a path which connects the objective part and a
                disassemblable part (parts) comes out, or
             2) no more branching is available

      - branch nodes of the current level to make the next level
         * stop branching a node
            1) if it is a disassemblable part, or
            2) if it appears elsewhere in an upper level as an indirect
               ancestor node
```

```
        * stop branching and cut the path which is related to the node
          that appears elsewhere in an upper level as a direct
          ancestor node
    end-do
end-algorithm
```

3. Group Disassembly (G.D.A)

A G.D.A means a disassembly of several parts at the same time in the same direction. G.D.A increases efficiency of disassembly because it can reduce the number of disassembly steps and the number of parts disassembled compared with I.D.A.

3.1. Carpenter's Approach for G.D.A

In order to reduce the number of disassembly steps and the number of parts disassembled in a part replacement, we propose a new concept called "carpenter's approach". The situation for the explanation of this concept is as follows: "A carpenter wants to cut a rectangular block out of a log (Figure 4). However, he/she finds that both ends of the log are damaged (shaded parts) and can not be used." In this situation, the carpenter will cut out the damaged part as much as possible with the least number of cuts. Due to this, the carpenter will take a cut at x rather than at y or at z to eliminate the damaged part 'c'.

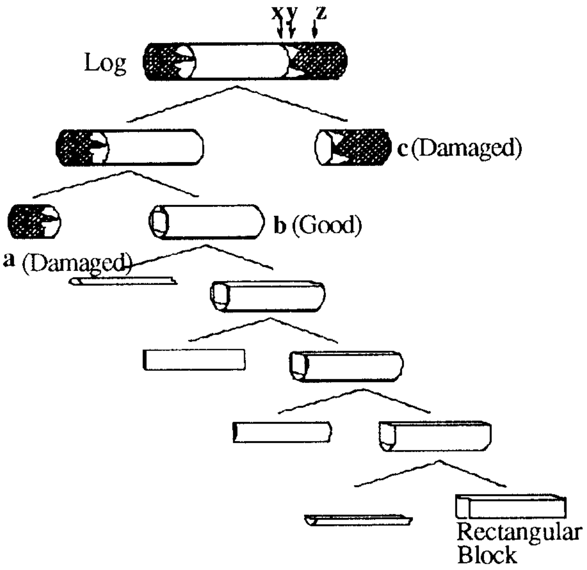


Figure 4. The Concept of Carpenter's Approach

The same concept will be applied to the disassembly problem. An assembly is divided into two groups: 1) the maximum group which does not include the objective part, and 2) the minimum group which includes the objective part. Subsequently, the minimum group is divided into two groups again. This grouping procedure continues until the objective part is disassemblable or no more grouping is available (algorithm: part replacement by G.D.A). The grouping structure is as shown in Figure 5.

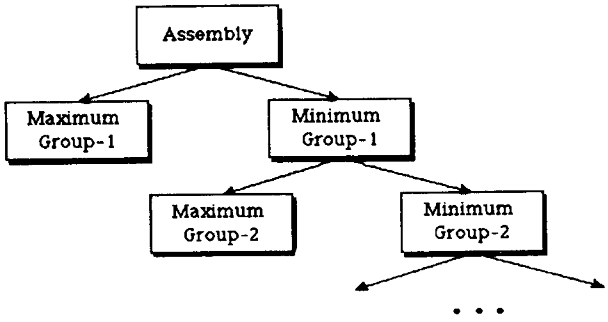


Figure 5. The Grouping Structure

The algorithm for part replacement by G.D.A is as follows:

```
algorithm  part replacement by G.D.A
   do-until  1) the objective part is disassemblable( ' • ' ), or
             2) no more grouping is available
        - step 1
            * neglect all cells in the column of the objective part
            * neglect all cells which include the objective part
        - step 2
            * for each row of direction, neglect all cells which
              include any part neglected in that row
        - step 3
            * for every direction, make a candidate group which
              consists of all parts not neglected
            * the maximum group is a group which has maximum
              number of parts (most out of candidate groups)
            * the minimum group is a group which consists of all
              parts which do not belong to the maximum group
            * the direction of the minimum group is opposite to the
              direction of the maximum group
        - step 4
            * let the minimum group be the assembly
   end-do
end-algorithm
```

An example assembly is given in Figure 6. We want to disassemble the objective part 'e' in order to replace it with a new part. The G.D.A concept will be applied for this assembly.

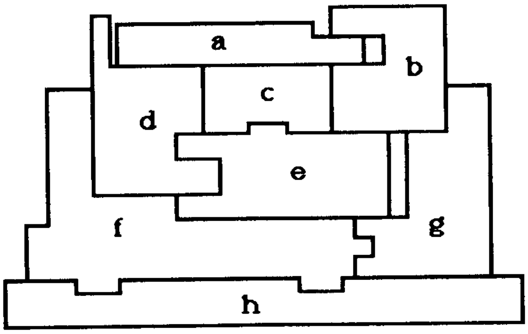


Figure 6. An Example Assembly for G.D.A

The grouping and disassembling procedure through the algorithm is summarized in Figure 7.

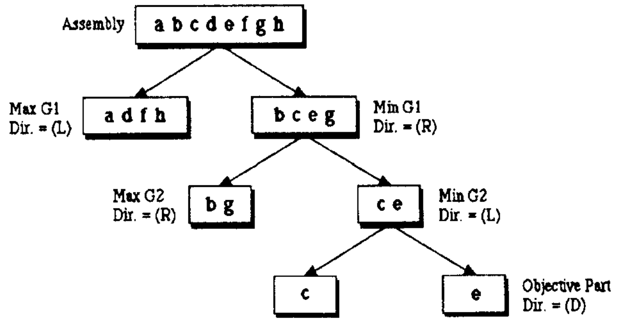


Figure 7. A Sequence for Part 'e' by G.D.A

We can disassemble the objective part 'e' through 3 steps; disassembling 4 parts (b,c,e, and g). If we follow the I.D.A procedure discussed before, one of the best sequences will be (g -> b -> a -> c -> e). That is, we have to disassemble at least 5 parts through 5 steps. Therefore, part replacement by G.D.A provides efficiency in terms of the number of disassembly steps and the number of parts disassembled.

In terms of the number of units of parts handled, however, 7 units (i.e, bceg + ce + e) are handled in the sequence by G.D.A. Compared with the number of units by I.D.A, G.D.A requires more handlings by 2 units. We will try to solve the handling problem again through the carpenter's approach.

3.2. Reducing the Number of Handlings

If the damaged part 'c' is bigger (or heavier) than the part 'a + b' (i.e, if c > a + b) in the log cutting problem, the carpenter will take a cut to discard the part 'c' and then he/she will transport only the part 'a + b' to another workplace rather than taking away the part 'c' because of ease of handling. However, as the part 'a' is also damaged, he/she will take a cut to discard the part 'a' and transport only the part 'b' to the workplace. This will reduce the material handling effort. This concept is applied to the disassembly problems to reduce the number of units of handling. If (Max G_{i+1}) has any common disassembly direction with (Min G_i), then the grouping structure can be modified as in Figure 8.

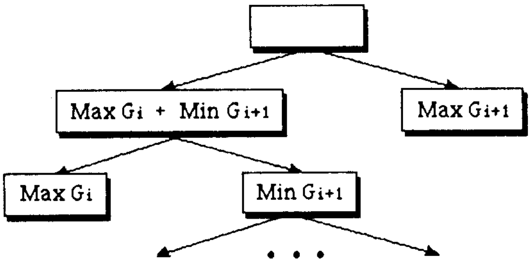


Figure 8. The Modified Grouping Structure

In our example, the objective part 'e' is disassembled through 3 steps and 5 units of handling according to this concept.

IV. IMPLEMENTATION AND AN ILLUSTRATIVE EXAMPLE

The approach proposed in this paper has been implemented in LISP on a MAC IIcx PC system. The system has been tested for various assemblies ranging from simple ones to more complex ones in terms of the number of parts and of the relationships of the parts. Figure 9 shows one of the assemblies. It is a toner cartridge in a laser printer which consists of several dozens of parts. In this study, it is disassembled to have 31 parts as shown in the figure. Some assumptions are followed as it was not designed with the concept of 'Design for Assembly'. One of the assumptions is such that the screws s1 and s2 are not considered because they require one additional assembly/disassembly direction for screwing them (part 20 could be fixed without them through the design change). Input data for this assembly can be easily prepared even though it is a fairly complex assembly in terms of assembly/disassembly sequence generation because it requires considerations of the relationship between only two parts at a time. A part of the data (B Matrix) is shown in Figure 10. Based on this data the system generates C matrix and Sweeping Table, and gives options of sequence generations; 1) F.D.A, 2) I.D.A, 3) G.D.A, and 4) M-G.D.A (Modified Group Disassembly). The output of the example assembly is partially shown in Figure 11. It shows disassembly sequences and disassemblable directions. The order of disassemblies of parts in a particular step in the case of F.D.A can be exchanged. For every step, disassembly can be implemented in two ways. At step 2 in F.D.A, for example, part 6 can be disassembled by; 1) disassembling part 6 from the subassembly (part 7, . . . , part 31) in the direction of D, or 2) disassembling the subassembly (part 7, . . . , part 31) from part 6 in the direction of U. The selection depends on the situation, where we need the concept of 'Design for Assembly'.

In F.D.A of our example, parts of the module-1 are disassembled one by one first (step 1 and 2), five screws, a cap, and a grip ring are disassembled (step 3), and then parts of module-2, and module-3 are disassembled (through step 4 to 12). This assembly is disassemblable thoroughly by means of one by one disassembly as shown in F.D.A. If this is infeasible, then the F.D.A process would stop. Therefore, this system could be utilized for the sensitivity analysis in order to identify which parts should be re-designed for assembly or disassembly automation. For a new design after some modification, preparing the input data for the analysis is very simple. If part i is re-designed, for example, the changes in the input data happen only in the column and row of the part i in the B matrix.

The objective part in G.D.A is a magnetic bar numbered as part 17. It is to be replaced with a new one. Through step 1 and 2, the module-1 except part 6 is disassembled. At step 3 module-3 is disassembled by module instead of by part. The effect
of G.D.A compared to I.D.A is mainly reflected in this step. Now module-2 is placed on the part 6. Module-2 is taken in the direction of U at step 4. At step 5 the system knows that disassembling parts (7 9 10 16 17) in the direction of L is better than disassembling parts (8 12 13 14 15 17 18 19) in the direction of R. Step 6 says that the objective part 17 can be taken from the group (7 9 10 16 17) directly. Therefore six steps are required for the
part 17 by G.D.A. Meanwhile, more number of steps will be required if I.D.A is used. Grouping is important because not only it minimizes the required number of steps but also it minimizes the number of changeovers of the assembly or disassembly directions because parts in a group can be disassembled in the same direction. Step 6 is not necessary as part 17 is taken out during the grouping of parts (7 9 10 16 17). This is explained by M-G.D.A which reduces the number of parts handling to 21 units.

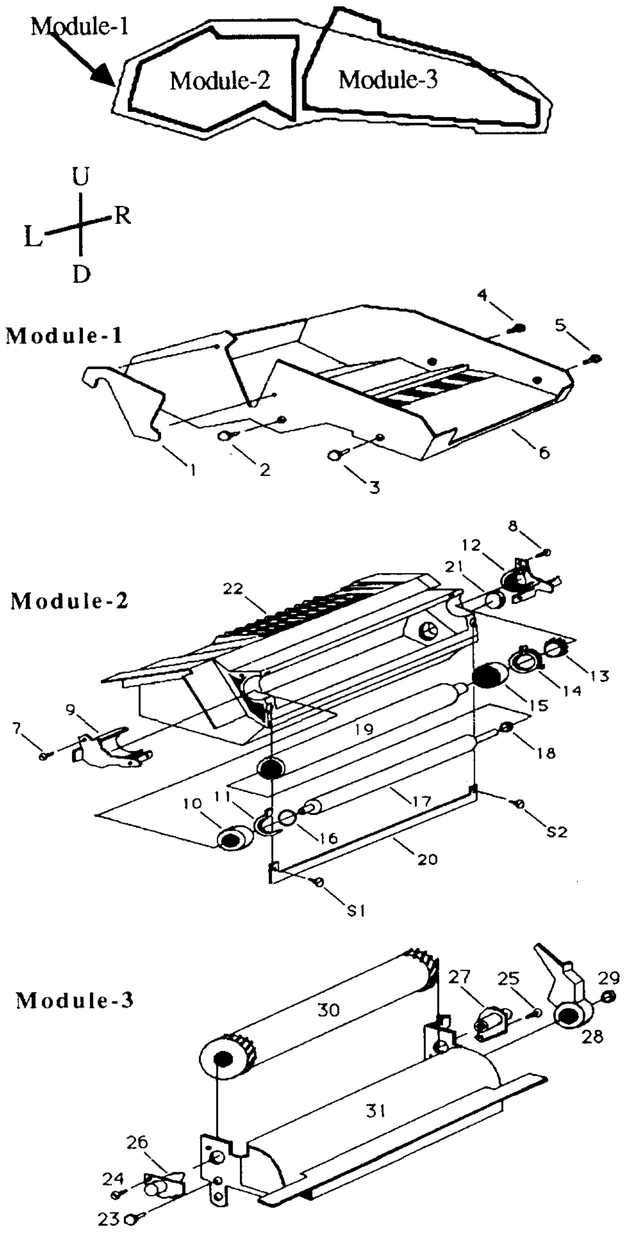


Figure 9. Toner Cartridge with 31 parts

```
; "Toner Cartridge"
(setq B '(
     (    )(L U R )(L U R D)(L U R D)(L U R D) (   R )( U R D)( U R D)( U R D)( U R D)
     ( U R D)( U R D)( U R D)( U R D)( U R D) ( U R D)( U R D)( U R D)( U R D)( U R D)
     (L U R D)( U R D)(L U R D)(L U R D)(L U R D) (L U R D)(L U R D)(L U R D)(L U R D)(L U R D)( U R D)

     (L  R D)(    )(L U R D)( U R D)(L U R D) (   R )(L U R D)(L U R D)( U R D)(L U R D)
     (L U R D)( U R D)(L U R D)(L U R D)(L U R D) (L U R D)(L U R D)(L U R D)(L U R D)(L U R D)
     (L U R D)(L U R D)(L U R D)(L U R D)(L U R D) (L U R D)(L U R D)(L U R D)(L U R D)(L U R D)(   R )
       • • •
```

Figure 10. Input Data for Toner Cartridge

```
-----------------------------------------------------------------------------------
                  M a i n     M e n u
===================================================================================
             1.        F.D.A - Full  Disassembly
             2.        I.D.A - Individual Disassembly
             3.        G.D.A - Group Disassembly
             4.        M-G.D.A - Modified Group Disassembly
             5.        Help
             6.        Stop
-----------------------------------------------------------------------------------

 =>  Select please  1

*********************************************************
*                 F u l l    D i s a s s e m b l y                *
*********************************************************
  * Step 1.  Disassemble  ((1 L) (2 L) (3 L) (4 R) (5 R))
  * Step 2.  Disassemble  ((6 D))
  * Step 3.  Disassemble  ((7 L) (8 R) (21 R) (23 L) (24 L) (25 R) (29 R))
  * Step 4.  Disassemble  ((26 L) (28 R))
  * Step 5.  Disassemble  ((27 R))
  * Step 6.  Disassemble  ((30 U))
  * Step 7.  Disassemble  ((31 U))
  * Step 8.  Disassemble  ((9 L) (12 R) (20 D))
  * Step 9.  Disassemble  ((10 L) (13 R))
  * Step 10.  Disassemble  ((11 L) (14 R) (16 L) (22 L))
  * Step 11.  Disassemble  ((15 R) (17 L))
  * Step 12.  Disassemble  ((18 L) (19 R))
-----------------------------------------------------------------------------------
                  F i n i s h e d    F.D.A
-----------------------------------------------------------------------------------
• • •
 =>  Input your OBJECTIVE part number  17

*********************************************************
*               Part    Replacement   by    G.D.A               *
*********************************************************
  * Step 1.   Disassemble  into  Two  Groups

 Group 1  =>  (1 2 3)  in the direction of L
 Group 2  =>  (4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
              23 24 25 26 27 28 29 30 31)  in the direction of R
-----------------------------------------------------------------------------------
  * Step 2.   Disassemble  into  Two  Groups

 Group 1  =>  (4 5)  in the direction of R
 Group 2  =>  (6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
             24 25 26 27 28 29 30 31)  in the direction of L
-----------------------------------------------------------------------------------
  * Step 3.   Disassemble  into  Two  Groups

 Group 1  =>  (23 24 25 26 27 28 29 30 31)  in the direction of U
 Group 2  =>  (6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22)  in
              the direction of D
-----------------------------------------------------------------------------------
  * Step 4.   Disassemble  into  Two  Groups

 Group 1  =>  (6 20)  in the direction of D
```

```
Group 2  =>  (7 8 9 10 11 12 13 14 15 16 17 18 19 21 22)  in the
             direction of U
-------------------------------------------------------------------
 * Step 5.  Disassemble  into  Two  Groups

Group 1  =>  (8 11 12 13 14 15 18 19 21 22)  in the direction of R
Group 2  =>  (7 9 10 16 17)  in the direction of L
-------------------------------------------------------------------
 * Step 6.  Disassemble  into  Two  Groups

Group 1  =>  (7 9 10 16)  in the direction of L
Group 2  =>  (17)  in the direction of R
-------------------------------------------------------------------
                     F i n i s h e d    G.D.A
-------------------------------------------------------------------
```

Figure 11. Generated Sequences for Toner Cartridge

V. CONCLUSIONS

The generation of assembly or disassembly sequence is a complex process, because the number of alternatives grows exponentially as the number of parts increases. The reverse of the disassembly sequence is a valid assembly sequence under the assumption that no external or internal forces affect the parts. In this paper, a new approach and its implementation were described for the generation of disassembly sequences through "Freedom and Interference Spaces" for the cases of 1) F.D.A, 2) I.D.A, and 3) G.D.A. This new approach seems to be practically applicable in terms of the volume of input data, the easiness of getting the input data, and the straightforwardness of the algorithms. It requires n(n-1)/2 input data for an assembly with n parts. The proposed approach can handle sensitivity analysis of a given assembly.

REFERENCES

[1] De Fazio, T. L. and Whitney, D. E., "Simplified Generation of All Mechanical Assembly Sequences," IEEE J. of Robotics and Automation, Vol. RA-3/6, pp.640-658, 1987.

[2] Holbrook, A. E. K. and Sackett, P. J., "Design for Assembly - Guidelines for Product Design," Assembly Automation: Proc. of the 9th Int. Conf., pp. 201-212, Mar. 1988.

[3] Homem de Mello, L. S. and Sanderson, A. C., "AND/OR Graph Representation of Assembly Plans," Carnegie-Mellon Univ., The Robotic Institute, Pittsburgh, PA, Rep. CMU-RI-TR-86-8, Apr. 1986.

[4] Huang, Y. F. and Lee, C. S. G., "A Framework of Knowledge-Based Assembly Planning," Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation, Sacramento, California, pp. 599-604, Apr. 1991.

[5] Ko, H. and Lee, K., "Automatic Assembling Procedure Generation from Mating Conditions," Computer-Aided Design, Vol. 19/1, pp. 3-10, 1987.

[6] Lee, Young-Q, and Kumara, Soundar R. T., "Individual and Group Disassembly Sequence Generation through Freedom and Interference Spaces," to appear in Journal of Design and Manufacturing, 1992.

[7] Shpitalni, M., Elber, G., and Lenz, E., "Automatic Assembly of Three-Dimension Structures via Connectivity Graphs," Annals of CIRP, Vol. 38/1, pp. 25-28, 1988.
[8] Toussaint, G. T. and El Gindy, H. A., "Separation of Two Monotone Polygons in Linear Time," Robotica, Vol. 2, pp. 215-220, 1984.
[9] Wolter, J. D., "On the Automatic Generation of Plans for Mechanical Assembly," Ph.D thesis, Univ. of Michigan, Dept. of Computer, Information and Control Engineering, Sep. 1988.
[10] Woo, T. C., "Automatic Disassembly and Total Ordering in Three Dimensions," Winter Annual Meeting of ASME, PED Vol. 25, pp.291-303, 1987.

---

* This work is scheduled to be presented at the 24th CIRP International Seminar on Manufacturing Systems, Denmark, June 1992.