Sādhanā Vol. 41, No. 1, January 2016, pp. 67–74

Liaison concatenation – A method to obtain feasible assembly sequences from 3D-CAD product

M V A RAJU BAHUBALENDRUNI* and BIBHUTI BHUSAN BISWAL

Department of Industrial Design, National Institute of Technology, Rourkela 769008, India
e-mail: bahubalindruni@gmail.com; bbbiswal@nitrkl.ac.in

MS received 31 December 2013; revised 28 February 2015; accepted 10 July 2015

Abstract. Selection of optimized assembly sequence from the available feasible assembly sequences is significantly essential to achieve cost-effective manufacturing process. To achieve this, at the outset, generation of feasible assembly sequences with topological, geometrical, precedence and stability conditions should be generated. The increase of part count in a product results huge number of assembly sequences, the Liaison matrix/Liaison graph generated based on the connections between the assembly components eliminates non-possible assembly sequences at the initial phase. There exist methods namely cut-set method to eliminate the non-possible assembly sequences using liaison graph. In this paper a new approach to eliminate the non-possible assembly sequences based on liaisons is described and the correctness of the methodology is illustrated with an example. The methodology to obtain the feasible assembly sequences is also briefly described and presented. An algorithm to interface with the CAD environment is described briefly.

Keywords. Assembly sequence planning; liaison matrix; liaison concatenation; assembly automation.

1. Introduction

Optimized assembly sequence of a product always plays major role in manufacturing industry in terms of cost-effectiveness by optimizing the overall usage of machine tools and fixturing. It also supports in reducing the lead time of the product and improving the final product quality as well. Most of the researchers worked on finding out the at least one feasible solution to assemble the product, and to find out the optimized assembly sequence from the feasible and stable assembly sequences with considerable approximations and assumptions. Finding feasible and stable assembly sequences from all set of possible assembly sequences involves three major phases.

Phase-1 involved in eliminating the non-possible assembly sequences based on the connections between parts from all set of assembly sequences. There exist methods to represent all set of assembly sequences, which are AND/OR graph and Directed Graph [1–3] methods. The elimination of non-possible assembly sequences is based on the contacts between parts in the assembly. As the number of parts increases in an assembly, the all set of assembly sequences also increase exponentially. Hence this phase involves lot of computational time in eliminating the non-possible assembly sequences. The Liaison matrix/Liaison graph [4] is the simplest one to establish the connections between the parts. By employing Cut-set method [2, 3] on the Liaison graph, the elimination of non-feasible assembly sequences can be done.

Phase-2 involved in finding the feasible assembly sequences from the liaison based assembly sequences by querying for precedence order. The questions can be generated by a computer program automatically using the methods developed by either Bourjault [4], De Fazio & Whitney [5] or assembly cut-set queries by Homem de Mello & Sanderson [2, 3].

Phase-3 involved in finding out the stable assembly sequences from the feasible assembly sequences. At later phases, optimal stable and feasible assembly sequence can be obtained subjected to assembly cost or assembly time. The phases of assembly sequence planning and the methodologies, those can be used at different phases are listed in table 1.

This paper is more focused on the Phase-1 and II, and developed a new approach to obtain the possible assembly sequences by liaisons between the assembled parts and tested these sequences for further feasibility. The approach is tested on an assembled product through an interfacing program. Computer Aided Three dimensional Interactive Applications (CATIA) software is a user friendly CAD tool with the feasibility of automation through Visual Basic Scripting or CAT-Scripting. CATIA is used to create 3D product and Visual Basic (VB)-scripting is used to interface with CATIA V5 R17 and to obtain the liaison matrix.

*For correspondence

Table 1. Assembly sequence planning phases and methodologies.

Phase no.	Phases	Methodologies	Approach
1	Obtain liaison data for the assembly product	Liaison graph [4]/Liaison matrix [6]	Manual/automated (through CAD interface)
1	Find out possible assembly sequences based on liaisons	Cutset method [7]	Manual/automated (through CAD interface)
2	Get feasible assembly sequences based on precedence relations	Queries on liaisons [4]/ [5] Queries on assembly cut-sets [2, 3]	Manual/semi-automated (with user intervention)
3	Obtain stable assembly sequences based on stability criteria	Assembly joining methods and constraints	Manual/semi-automated (with user intervention)
4	Optimize for assembly cost (and/or) time	Multidisciplinary optimization algorithms	Automated

2. Liaison graph and liaison matrix

2.1 *Liaison graph*

Liaison diagram is a graphical representation of connections between the assembled parts of a product. Though this method is initialized by [4] and later it is popularized by De Fazio & Whitney [5]. A liaison is a defined connections established between the components. The liaison diagram typically consists of nodes and hyper-arcs connected to the nodes. The nodes represent the part, and the hyper-arcs from the node to other nodes represent the connections with the mating parts. These connections will be named to use further to establish precedence relations between the connections. A 4-part assembly shown in figure 1 is used to demonstrate the mechanism.

The liaisons graph for 4-part assembly shown in figure is represented in figure 2. The nodes of the liaisons graph indicate the parts and the hyper-arcs connected between the parts represent liaisons between the parts.

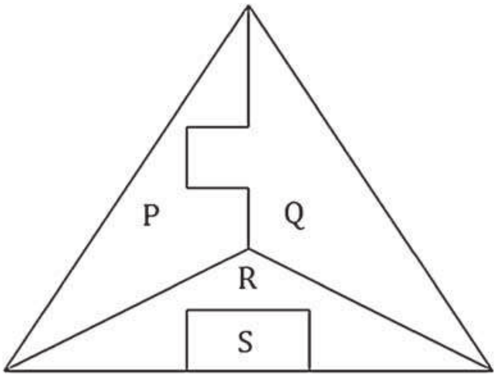


Figure 1. 4-Part assembled product.

2.2 *Liaison matrix*

Matrix representation of liaisons between the components with binary codes 1, 0 is proposed by [6]. A *n-by-n* matrix is required to represent the liaisons for a product created by "*n*" number of components. The diagonal elements of this matrix consist of null values, and the row of matrix represents the liaisons between one component and the other components in the assembly. The column of matrix represents the components connected by liaison relationships. The sub-matrices of *n-by-n* matrix represent the local liaison relationships in subassemblies. The elements of liaison matrix for the assembly shown in figure 1 are listed in table 2.

3. Liaison concatenation

Liaison concatenation technique works based on the liaisons between the components represented in the liaison matrix.

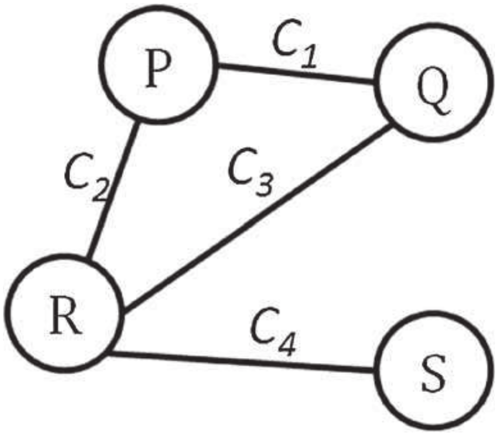


Figure 2. Liaison graph for 4-part assembly.

The mechanism takes data only once from the liaison matrix and develops assembly sequence by concatenating the liaisons. The Liaison concatenation first identifies the number of contacts (between two parts only) and stores in an array format (Pre array). The Pre array is of size 2k-by-2 for a product with "k" number of liaisons. The Pre array for the 4-Part assembly based on the liaisons is listed in table 3. The Pre array is considered as Post array at the initial stage.

In the next level of operation, if there exist any element of the Pre array set-1 as first element of Post array, then the second element of Post array will be concatenated to the Pre array set, provided that the second element of Post array does not exist in the Pre array set. Existence of second element of the Post array in the Pre array leads to no output (O/P). The level-3 liaison concatenation operation is shown in figure 3.

When an output set is created, it will be compared with the prior sets and eliminate if similar set exist. The resulted set is a new Pre array to carry out the next level concatenation. If the level value reaches to the total number of parts the Pre array represents the total possible assembly sequences by liaisons. Figure 4 represents the Final level of Liaison concatenation process.

Table 2. Four-part assembly liaison matrix elements.

	P	**Q**	**R**	**S**
P	0	1	1	0
Q	1	0	1	0
R	1	1	0	1
S	0	0	1	0

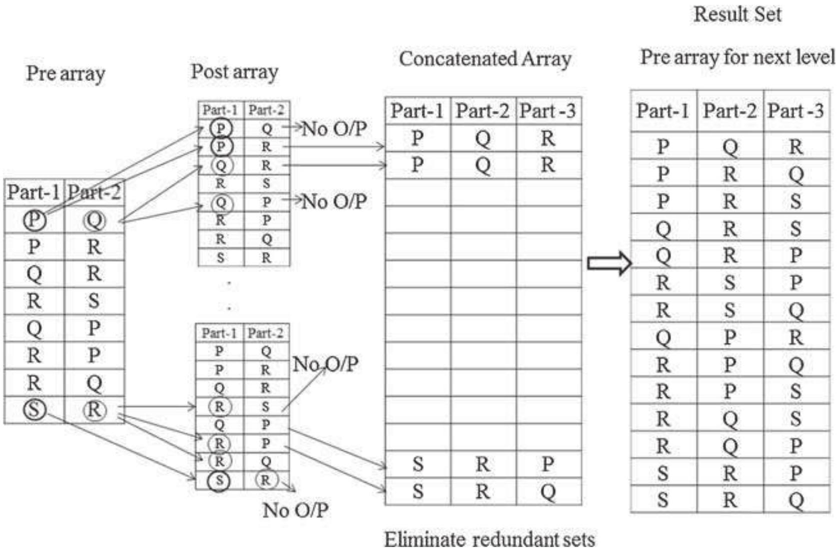
Table 3. Elements of Pre array of [2k][2].

i	Pre array[i][1]	Pre array[i][2]
1	P	Q
2	P	R
3	Q	R
4	R	S
5	Q	P
6	R	P
7	R	Q
8	S	R

4. Correctness of methodology

The correctness of the methodology is verified by assessing the Liaison concatenation method results with Cut-set method resulting assembly sequences. The main assumption in cut-set method is that, if a part can be disassembled from the product at a phase without any destructive operation, the part can be assembled at the same phase. The assembly cut-sets are generated manually from the liaison graph represented in figure 2. Table 4 lists the all possible assembly cut-sets for 4-part assembly. For the present case, generation of sub-assemblies is not considered and hence "can a sub assembly be removed" option is not listed in the table.

At level three each sub-graph can have two options either can part-1 be removed or part-2 be removed, one answer is valid for both the questions. The possible dis-assembly sequences from the assembly cut-sets and the reverse of the disassembly sequences are represented in table 5.

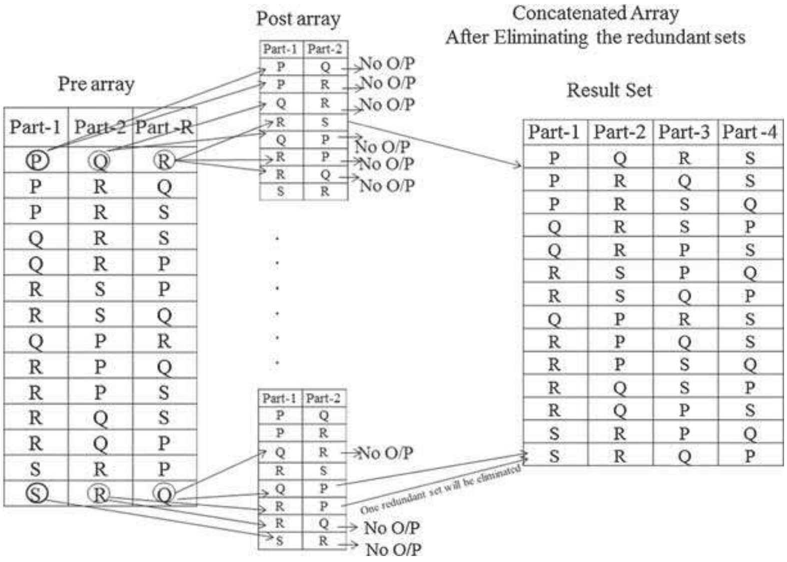


Part-1	Part-2
P	Q
P	R
Q	R
R	S
Q	P
R	P
R	Q
S	R

Part-1	Part-2	Part -3
P	Q	R
P	Q	R
S	R	P
S	R	Q

Part-1	Part-2	Part -3
P	Q	R
P	R	Q
P	R	S
Q	R	S
Q	R	P
R	S	P
R	S	Q
Q	P	R
R	P	Q
R	P	S
R	Q	S
R	Q	P
S	R	P
S	R	Q

Figure 3. Level-3 liaison concatenation process.


Post array
Concatenated Array
After Eliminating the redundant sets
Pre array
Result Set
No O/P
One redundant set will be eliminated

Pre array

Part-1	Part-2	Part -R
P	Q	R
P	R	Q
P	R	S
Q	R	S
Q	R	P
R	S	P
R	S	Q
Q	P	R
R	P	Q
R	P	S
R	Q	S
R	Q	P
S	R	P
S	R	Q

Post array

Part-1	Part-2
P	Q
P	R
Q	R
R	S
Q	P
R	P
R	Q
S	R

Part-1	Part-2
P	Q
P	R
Q	R
R	S
Q	P
R	P
R	Q
S	R

Result Set

Part-1	Part-2	Part-3	Part -4
P	Q	R	S
P	R	Q	S
P	R	S	Q
Q	R	S	P
Q	R	P	S
R	S	P	Q
R	S	Q	P
Q	P	R	S
R	P	Q	S
R	P	S	Q
R	Q	S	P
R	Q	P	S
S	R	P	Q
S	R	Q	P

Figure 4. Level-4 liaison concatenation process.

5. Feasible assembly sequences

The assembly sequences obtained based on the liaisons must be tested for possibility of assembling. Based on the assumption "If a part can be disassembled from the product at a phase without any destructive operation, the part can also be assembled at the same phase", feasibility testing will be done for each assembly sequence. The sequences must be placed in ascending/descending order based on the last part of the assembly in order to reduce the computational time. Table 6 indicates the list of liaison based assembly sequences in ascending order based on the last part.

A set of questions will be asked on these sequences, the first set of questions will be "can the last part be disassembled?". For the sequences (1–3) the last part is "P" and similarly the maximum extent "n−1" questions have to be answered, where "n" is the number of parts. The answer will be obtained by moving the part in +x, −x, +y, −y, +z and −z directions in the work space. If answer is found to be "YES" for any directions the remaining directions will not be checked. For 4-part assembled product shown in figure 1, the work space and centre of gravity is mentioned in figure 5.

For the present example, a 2D feasibility test is performed and shown in table 7, the centre of gravity values are used as reference to move the part in a specific direction and to perform the contact analysis to check the possibility of interference.

Table 4. Assembly cut-sets for 4-part assembly.

Level	Graph/sub graph	Can part-i be disassembled	Sub graph/ node
1	P,Q,R,S	P	Q,R,S
		Q	P,R,S
		S	P,Q,R
2	Q,R,S	Q	R,S
		S	Q,R
	P,R,S	P	R,S
		S	P,R
	P,Q,R	P	Q,R
		Q	P,R
		R	P,Q
3	R,S	R	S
	P,Q	P	Q
	P,R	P	R
	Q,R	Q	R

Table 5. Assembly sequences resulted by cut-set method.

S. no.	Disassembly sequences	Assembly sequences
1	P Q R S	S R Q P
2	P Q S R	R S Q P
3	P S Q R	R Q S P
4	P S R Q	Q R S P
5	Q P R S	S R P Q
6	Q P S R	R S P Q
7	Q S P R	R P S Q
8	Q S R P	P R S Q
9	S P Q R	R Q P S
10	S P R Q	Q R P S
11	S Q P R	R P Q S
12	S Q R P	P R Q S
13	S R P Q	Q P R S
14	S R Q P	P Q R S

Table 6. List of liaison based assembly sequences in ascending order.

S. no.	Disassembly sequences
1	Q S R P
2	S Q R P
3	S R Q P
4	P S R Q
5	S P R Q
6	S R P Q
7	P Q S R
8	P S Q R
9	Q P S R
10	Q S P R
11	S P Q R
12	S Q P R
13	P Q R S
14	Q P R S

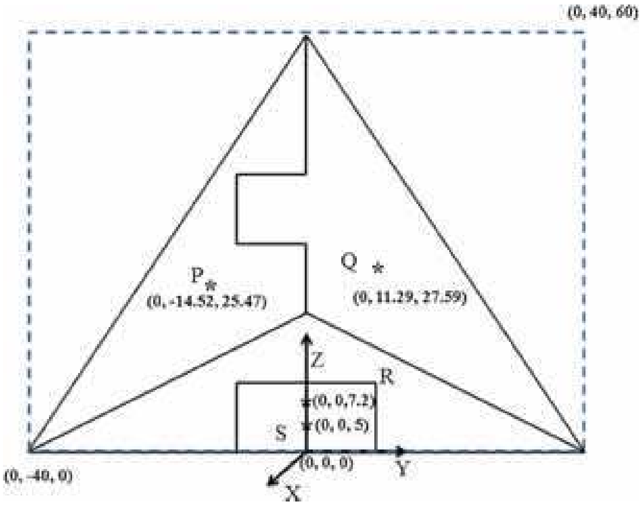


Figure 5. Work space and part level centre of gravity values.

Non-feasible assembly sequences are eliminated at each level of feasibility test and the qualified feasible assembly sequences at first level are sent for the next level feasibility test. The test will be done for n−2 times, where "n" is the number of parts. After n−2 feasibility tests, the resulted assembly sequences are completely feasible assembly sequences.

6. Algorithm and computer integration

The basic prerequisite to perform liaison concatenation method is Liaison graph/ Liaison matrix. In this section, an algorithm to extract the liaison matrix from 3D CAD product and algorithm for liaison concatenation using liaison matrix is explained.

6.1 *Liaison matrix extraction algorithm*

There exists any one of the three different types of conflicts between any two components in an assembled product. First type of conflict is clearance indicated by a positive conflict value represents gap between the two components. Second type of conflict is clash indicated by a negative conflict value represents the depth interference value between two components. If a negative conflict value exists, it indicates that neither the parts are assembled properly nor the parts are created correctly. In such cases, the liaison matrix extract code will pop-up an error and terminates the process. Third type of conflict is contact indicated by a null conflict value represents the contact between two or more faces of the components. Our interest is in capturing the pairs of parts which are in contact and placing them in the matrix format. The liaison extraction code is demonstrated below and flowchart is represented in figure 6.

Table 7. Level-1 feasibility test for 4-part assembly.

Assembly sequence	Feasibility test question	+ Y	− Y	+ Z	− Z	Sequences for next level
Q S R P	Can part "P" Disassembled	No	Yes			QSR
S Q R P			Yes			SQR
S R Q P			Yes			SRQ
P S R Q	Can part "Q" Disassembled	Yes				PSR
S P R Q		Yes				SPR
S R P Q		Yes				SRP
P Q S R	Can part "R" Disassembled	No	No	No	No	
P S Q R		No	No	No	No	
Q P S R		No	No	No	No	
Q S P R		No	No	No	No	
S P Q R		No	No	No	No	
S Q P R		No	No	No	No	
P Q R S	Can part "S" Disassembled	No	No	No	Yes	PQR
Q P R S					Yes	QPR

Liaison extraction code

```
1. Open an assembly in CATIA Assembly design
2. Obtain the number of parts in the Assembly "say n number of parts"
3. Create a null matrix of "n-by-n"
4. Compute the conflicts using "Clash computation type between all"
5. Obtain total number of conflicts "m"
6. For each conflict 1 to m
     Define the conflict type by conflict value
     If Conflict Value <0
       Exit from the loop and pop up and error message
     If Conflict Value = 0
       Identify the conflict product.1 name in the parts list say i^th part
       Identify the conflict product.2 name in the parts list say j^th part
     Replace the null value with "1" for the [i][j] and [j][i] elements of null matrix
7. Export the matrix data to Ms-Excel
```

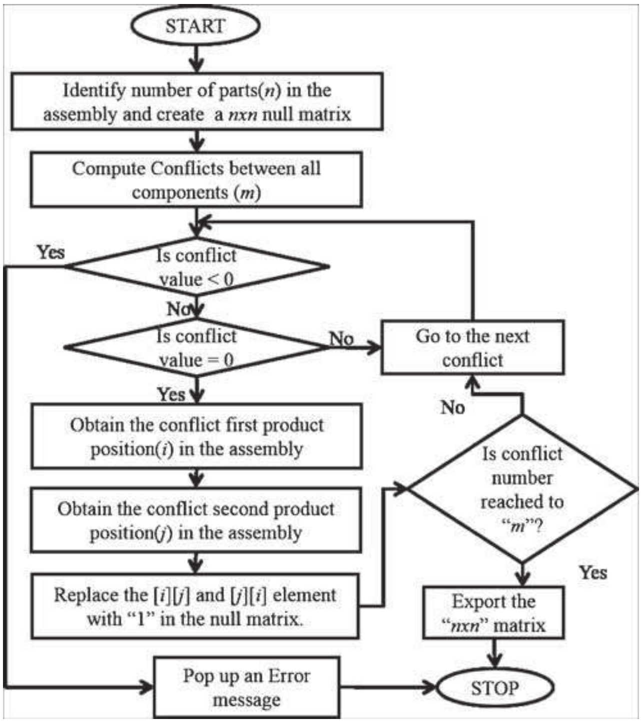

START
Identify number of parts(n) in the assembly and create a nxn null matrix
Compute Conflicts between all components (m)
Is conflict value < 0
Yes
No
Is conflict value = 0
No
Yes
Go to the next conflict
Obtain the conflict first product position(i) in the assembly
Obtain the conflict second product position(j) in the assembly
Replace the [i][j] and [j][i] element with "1" in the null matrix.
Is conflict number reached to "m"?
No
Yes
Export the "nxn" matrix
Pop up an Error message
STOP

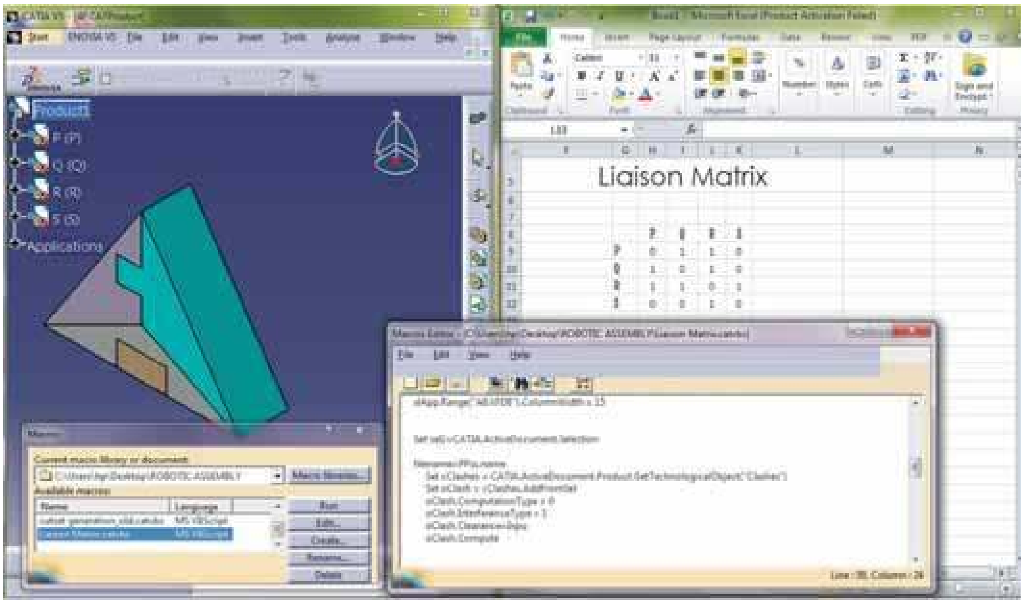
Figure 6. Liaison matrix extraction flowchart.

By executing the macro, the liaison matrix for the assembly will be generated and exported to Ms-excel as shown in figure 7.

The non-diagonal unit values of the matrix represent that there is contact between the parts indicated in row and column.

6.2 *Liaison concatenation algorithm*

The liaison matrix generated from the previous section will be considered as input for liaison concatenation algorithm. This method needs three two dimensional arrays, typically named as Pre, Post and Temp, in which Post array is a constant array throughout the algorithm. The second dimension of the Post array is 2, the first dimension of the Post array is equal to the number of "1"s in the Liaison matrix. The



Liaison Matrix

	P	Q	R	S
P	0	1	1	0
Q	1	0	1	0
R	1	1	0	1
S	0	0	1	0

```
xlApp.Range("A8:XFD8").ColumnWidth = 15

Set selJ = CATIA.ActiveDocument.Selection

Filename = PPlu.name
    Set cClashes = CATIA.ActiveDocument.Product.GetTechnologicalObject("Clashes")
    Set oClash = cClashes.AddFromSel
    oClash.ComputationType = 0
    oClash.InterferenceType = 1
    oClash.Clearance=0
    oClash.Compute
```

Figure 7. Macro interface with CATIA and Ms-Excel.

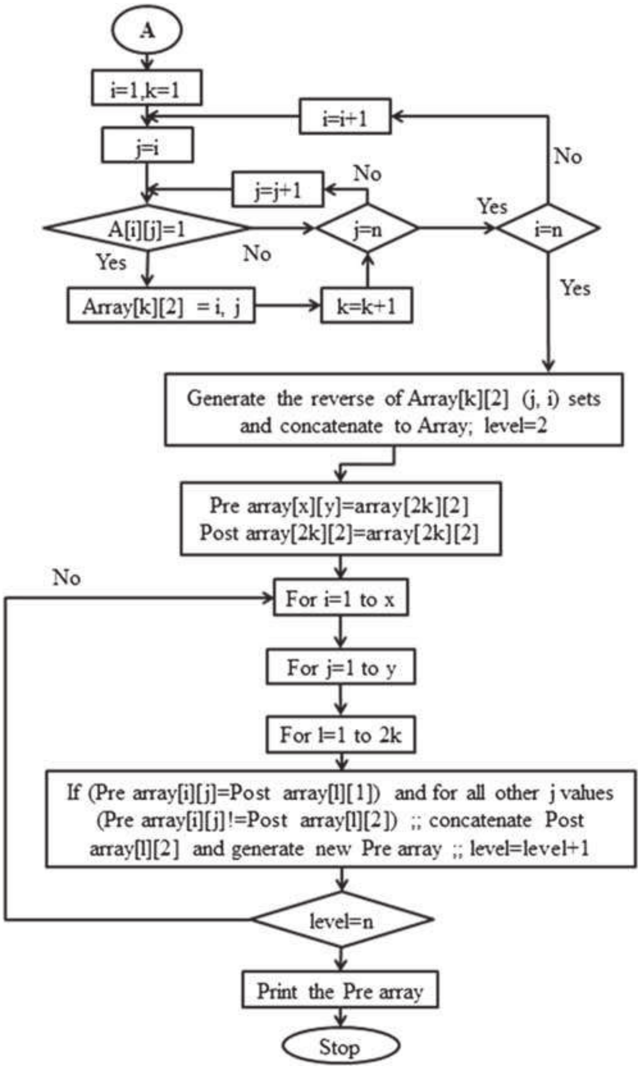

A
i=1,k=1
i=i+1
j=i
No
j=j+1
No
Yes
A[i][j]=1
j=n
i=n
Yes
No
Array[k][2] = i, j
k=k+1
Yes
Generate the reverse of Array[k][2] (j, i) sets and concatenate to Array; level=2
Pre array[x][y]=array[2k][2]
Post array[2k][2]=array[2k][2]
No
For i=1 to x
For j=1 to y
For l=1 to 2k
If (Pre array[i][j]=Post array[l][1]) and for all other j values (Pre array[i][j]!=Post array[l][2]) ;; concatenate Post array[l][2] and generate new Pre array ;; level=level+1
level=n
Print the Pre array
Stop

Figure 8. Liaison concatenation mechanism flow chart.

Pre array and Post arrays are of same size at the initial level and the Pre array size changes for each iteration. The temp array is the concatenated array of Pre and Post arrays, which will be treated as Pre array for the next iteration. A typical liaison concatenation mechanism is represented in figure 8.

The working mechanism of Liaison concatenation code is demonstrated below.

```
1. Obtain liaison matrix from CATIA
2. Set level=2
3. For each "i" value 1 to n (n is number of parts)
      For each "j" value i to n (n is number of parts)
      If a[i][j] is unity value then store i,j values in Pre
   array[k][1]=i; array[k][2]=j;
4. For each "i" value 1 to k (k is the length of Pre array)
      Pre array[k+i][1]= Pre array[i][2], Pre
   array[k+i][2]= Pre array[i][1].
      Post array[i][1]= Pre array[i][1], Post array[i][2]= Pre
   array[i][2].
      Post array[k+i][1]= Pre array[i][2], Post
   array[k+i][2]= Pre array[i][1].
5. For each "i" value 1 to x (Pre array is a 2 dimensional
   array of [x][y])
      For each "j" value 1 to y
      For each "l" value 1 to k (Post array is a 2 dimensional
   array of [2k][2])
      If (Pre array[i][j]=Post array[l][1]) & for all other j
   values (Pre array[i][j]!=Post array[l][2])
      Concatenate Post array[l][2] and generate new Pre
   array
6. The new Pre array size will be changed
      level=level+1
7. If level reaches to "n" export the Pre array else go to Step-
   4.
8. Stop
```

The proposed mechanism outcomes the resulted set represented in figure 4 for 4-part assembly.

7. Conclusions

In this paper, a new approach named "Liaison Concatenation" is proposed and illustrated with an example. The correctness of the proposed method is verified and presented. The mechanism of Liaison matrix extraction and integration of Liaison concatenation methodology on CATIA V5 tool is presented. The present method is very simple to implement and does not require any prior knowledge on the assembly to perform the concatenation operations. The automated process of liaison matrix extraction from CATIA and implementation of Liaison Concatenation method saves lot of computation time and it also avoids human errors to a great extent in extracting liaison data.

References

[1] Homem de Mello L S and Sanderson A C 1986 AND/OR graph representation of assembly plans. In: Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Morgan Kaufmann Publishers, 1113–1119

[2] Homem de Mello L S and Sanderson A C 1988a *Automatic generation of mechanical assembly sequences*. Technical Report, CMU-RI-TR-88-19, The Robotics Institute, Carnegie-Mellon University

[3] Homem de Mello L S and Sanderson A C 1988b Task sequence planning for assembly. In: IMACS World Congress '88 on Scientific Computation, Paris

[4] Bourjault A 1984 Contribution a uneApprocheMethodologique de L'AssemblageAutomatise: Elaboration Automatique des Sequences Operatoires ("Contribution to the

Methodology of Automated Assembly: Automatic Generation of Operations Sequences”), Ph.D. thesis, Université de Franche-Comté, Besançon, France (in French)

[5] De Fazio T L and Whitney D E 1987 Simplified generation of all mechanical assembly sequences. *IEEE J. Robotics Autom.* 3(6): 640–658

[6] Dini G and Santochi M 1992 Automated sequencing and subassembly detection in assembly planning. *CIRP Annals – Manuf. Technol.* 41(1): 1–4

[7] Homem de Mello L S 1991 A correct and complete algorithm for the generation of mechanical assembly sequences. *IEEE Trans. Robotics Autom.* 7(2): 228–240