Optimization of Automobile Assembly Process to Reduce Assembly Time

Neeraj Panhalkar[1], Ratnadeep Paul[2] and Sam Anand[3]

[1]University of Cincinnati, panhalnm@mail.uc.edu
[2]University of Cincinnati, ratnadeep.paul@uc.edu
[3]University of Cincinnati, sam.anand@uc.edu

ABSTRACT

The automobile manufacturing industry is undergoing a significant restructuring. Every automaker is investing heavily for adapting new manufacturing processes as well as assembly techniques which will reduce the overall operating costs while consistently maintaining the quality of the product. In general, assembly lines are being designed and developed with a goal to synchronize workers, machines, tools and components. It is important to have an optimized assembly operation sequence in order to facilitate the smooth functioning of an assembly line. This research paper will focus on developing an optimal vehicle assembly sequence based on the liaisons and precedence graphs concepts [3, 5]. In general, a liaison establishes the connectivity between different components in an assembly based on their connectivity with respect to each other. Using the existing liaisons, multiple feasible precedence graphs can be generated which in turn lead to the development of multiple feasible assembly sequences. These feasible assembly sequences can have different assembly times. From these multiple sequences, an optimized vehicle assembly sequence will be developed for achieving minimum assembly time. A dynamic tree based algorithm has been developed to automatically traverse the feasible liaison and precedence graphs in order to calculate the optimal assembly precedence sequence for minimum assembly time.

Keywords: assembly, optimization, assembly duration.

1. INTRODUCTION

Assembly is the process of joining two or more components, subassemblies or groups together to serve a desired functionality. In general, there are multiple sequences available to complete a particular task from start to end but not all sequences are feasible as they cannot satisfy the given set of constraints. It is important to select the optimal feasible sequence for an assembly to reduce factors affecting the overall cost. The factors such as total assembly time, number of workers required to complete the task and inventory cost of the components etc. affect the overall assembly cost. The order in which the components are assembled together is important in case of optimizing the assembly sequence. It is evident that out of all the possible assembly sequences some sequences are more effective than others in utilizing the available resources including workers, equipment and tools optimally.

Every assembly task can be divided into multiple subtasks wherein two or more components or modules will be joined together at each step of the task. This process involves establishing liaisons between individual components and generating feasible precedence graphs based on the liaison constraints [3,5]. The generated feasible sequences will make sure that the assembly of the product will perform as required, meeting all the functional requirements. To calculate the time associated with each of the precedence graphs, a pre-determined motion time systems like MOST (Maynard Operation Sequence Technique) can be used [14], [10], [11] & [13].

The overall cost and effectiveness of an assembly is greatly affected by the selection of an assembly sequence. Different assembly sequences may require different machines and tools leading to an altogether new plant setup with a different approach of utilizing the available resources. Typically, the assembly planning process is heavily dependent on the expertise and intuition of the industrial and manufacturing engineers. Though it is possible to come up with a feasible sequence using the experience of an engineer, the generated assembly sequence may not be the optimal sequence. Thus it is very important to generate all

Computer-Aided Design & Applications, 11(S1), 2014, S54–S60, http://dx.doi.org/10.1080/16864360.2014.914410

the feasible assembly sequences systematically without neglecting any of the feasible sequences thus ensuring the smooth functioning of the assembly process and utilizing the resources in an optimal way.

This paper concentrates on a methodology to generate feasible precedence sequence that will reduce the overall assembly time. The target market for this paper is an assembly shop which assembles customized cars satisfying the needs of the customers. Subsequently, a robust optimization model will be developed for a large scale assembly line used for mass production of vehicles. The next section will describe the methodology to select the optimal sequence among all the available feasible sequences that will reduce the overall assembly time.

2. LITERATURE REVIEW

Bourjault [3] in his research represented and modeled assembly constraints by using the knowledge about the liaisons between the components through a question and answer format. As the method of using the question and answer format proposed by Bourjault [3] was a cumbersome task, DeFazio and Whitney [5] developed a "diamond graph" search method to generate all assembly sequences. To further reduce the number of nodes in "diamond graphs", Homem de Mello and Sanderson [8] used AND/OR graph to solve sequence generation problem by using the disassembly approach. In the disassembly approach, the main idea is to list all possible disassembly ways by studying the liaison cut-sets of the liaison graph. The liaison graph is an undirected graph showing the connection between two parts represented by a node. As the process of generating all the possible assembly sequences requires large amount of input, manual sequence generation approaches fail to work for complex assembly structures.

Baldwin et al. [1] introduced an integrated computer aid to generate and evaluate the precedence sequences. Romney et al. [15] and Kaufman et al. [12] proposed an automated system based on the collision analysis. Bonneville et al. [2] proposed the use of genetic algorithms in addressing the issue of generating feasible sequences for large assemblies. Chen and Henrioud [4] proposed an algorithm to generate all feasible precedence sequences for an assembly based on the prior precedence information of the components. Tsao and Wolter [16] and Huang and Lee [9] introduced a sequence generation technique using predicate calculus methods. Delchambre and Wafflard [6] developed a software which extracts precedence constraints from a liaison graph. Delchambre and Gaspart [7] developed a prototype software for generation and selection of assembly sequences. For an automobile assembly plant, the precedence knowledge is very helpful if a new vehicle is to be assembled by modifying the existing line.

3. METHODOLOGY

It is very important to select the optimal feasible assembly sequence as it impacts critical factors such as duration of the total assembly, crew size required to assemble the product, specialized tools and fixtures requirement and also the amount of rework involved in an assembly. Generating all the feasible assembly sequences for a particular product can be very intensive for a product with large number of parts. As the number of parts in an assembly increases, the number of feasible sequences increases dramatically. Therefore, it becomes necessary to develop a method for generating feasible sequences automatically. Liaison graphs [3,5] and precedence diagrams [3] have been developed by researchers as a way for performing these tasks. In this paper, these two techniques have been used to develop a dynamic tree based methodology for generating an optimal assembly sequence for an automobile in order to minimize the total assembly time (Fig. 1). Before this methodology is explained in detail, the concept of liaison and precedence graphs has been explained in the following sub-section.

3.1. Liaison and Precedence Graphs

Bourjault [3] introduced a method to generate all feasible sequences algorithmically for an assembly by answering a set of questions. The questions revolved around the information about the mating of the parts. In general, a relationship graph between individual parts of an assembly is developed based on the mating conditions of the parts. Each part of the assembly is represented as a node of the graph and the line joining the two parts shows the connection between them. A user can extract the information out of the liaison graph representing the connection between parts of an assembly and an assembly sequence can be generated. Though the answers to the questions generate all feasible assembly sequences, it is difficult to follow the question-answer format for a large product with several thousand parts.

Defazio and Whitney [5] introduced a liaison graph generation method based on Bourjault's [3] work. In their approach a set of two questions must be answered for all liaisons. The questions are:

1) "*what liaison must be done prior to doing liaison i?*" [5]
2) "*what liaisons must be left to be done after doing liaison i?*" [5]

Fig. 2(a) and Fig. 2(b) shows the pictorial and graphical representation of a door sub-assembly respectively. The door subassembly has a total of 8 parts to be assembled together to complete the subassembly. The numbers in the parenthesis next to the part name in Fig. 2(b) are same as indicated in the Fig. 2(a).

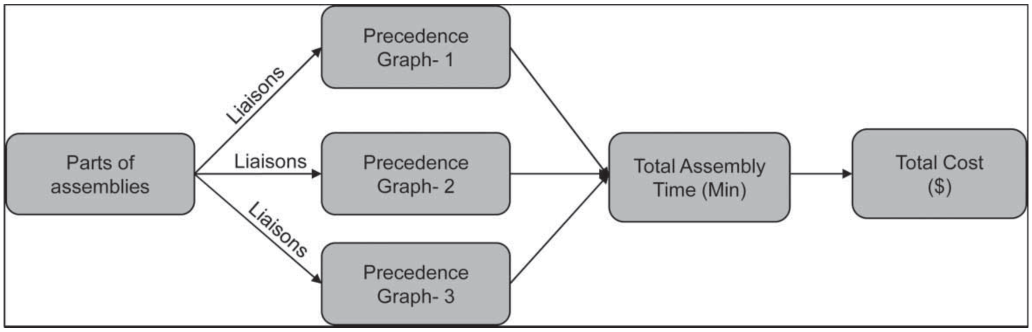

Fig. 1: Overview of the approach.

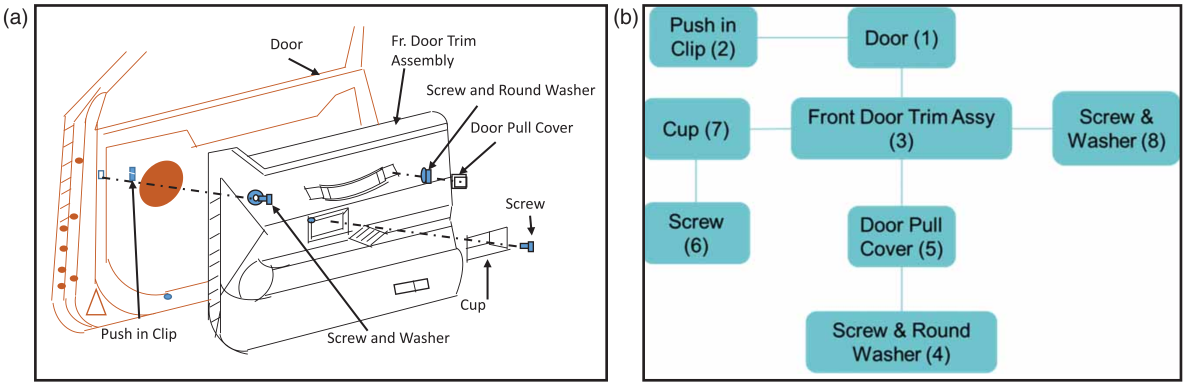

Fig. 2: (a) Pictorial representation of door sub-assembly [18] (b) Graphical representation of door sub-assembly.

Fig. 3 shows the liaison graph for the door sub-assembly using Defazio and Whitney's [5] method by answering the two basic questions in their paper. From the liaison diagram it can be seen that there are multiple ways to assemble the door assembly with a different starting point. One of the possible sequences may start with the door (part#1) followed by mounting the fully assembled front door trim assembly (part#3) on to the door. Another option would be to assemble the front door trim assembly on to the door and then installing the cup on the front door trim assembly. All these feasible sequences for the door assembly form a precedence graph as shown in the Fig. 4.

As shown in the precedence graph, at each level (level *i*) a worker has multiple options to move to the next level (level $i+1$). To find the optimal feasible assembly sequence for the door sub-assembly, a filtering criterion has to be defined to measure the efficiency of each of the feasible assembly sequence. For example, if the overall goal is to reduce the assembly time, MOST [14] analysis can be performed for each of the feasible sequence and the sequence with the least assembly time can be selected as the optimal feasible assembly sequence. As this paper tries to reduce the assembly time, an optimization model using the depth first graph search algorithm has been proposed. The advantage of using depth first search algorithm in conjunction with the precedence graph is to optimize the generation assembly sequence as we move on from level *i* to level $i+1$ in the precedence graphs.

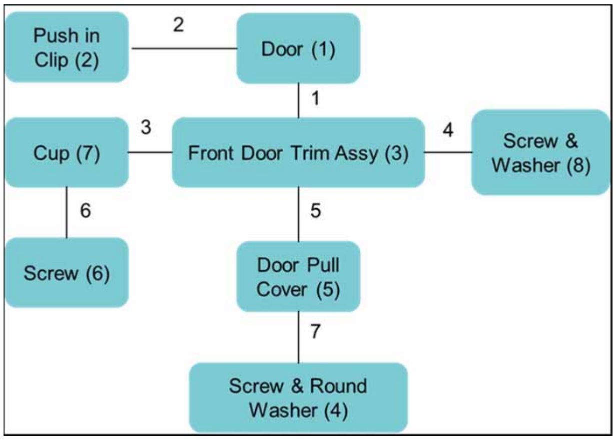

Fig. 3: Liaison diagram of the door sub-assembly.

3.2. Maynard Operation Sequence Technique and Calculation of Total Assembly Time

Maynard Operation Sequence Technique, commonly known as MOST is an industry accepted time measurement practice and is developed by H. B. Maynard and Company, Inc. [14]. MOST technique uses the knowledge about the series of activities involved

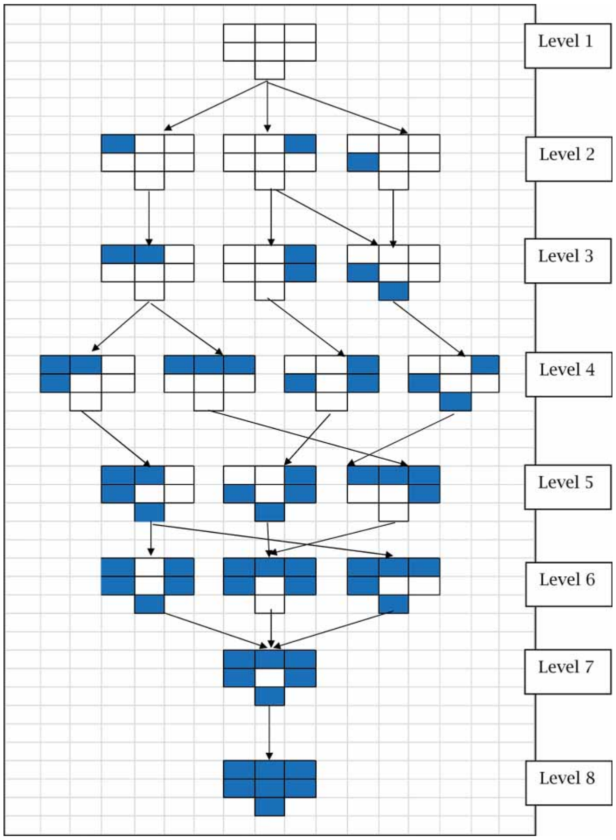

Fig. 4: Precedence graph for door sub-assembly.

while performing a task and calculates the estimated total time required to finish the task. MOST technique has been applied to a variety of manufacturing and industrial domains including automotive, aerospace and electronics industries. This easy to use tool is preferred by industries which enables them to managing their resources in an efficient manner, streamline the overall process under consideration, facilitate planning activities and also estimate the labor cost.

There are three different versions of MOST: MiniMOST, BasicMOST and MaxiMOST [19]. In general, the task is broken down into its multiple sub-tasks and time required to complete each of these sub-tasks is calculated. The estimated total time is obtained by adding all the individual times of these sub-tasks involved. The time calculated is expressed in terms of time measurement units (TMU). 1 TMU is equivalent to 1/100,000 of an hour or 0.0006 minutes [14]. The main difference amongst these versions of MOST is the level of detailing. If the activities involved are in the level of tens of TMUs, BasicMOST is used. The maxiMOST version works for task wherein the level of activities recorded are in hundreds of TMUs and if individual TMUs are to be used then miniMOST is preferred.

Tab. 1 is used to illustrate the MOST concept for a small activity "*Pick nut, fasten on bolt with 7 actions*".

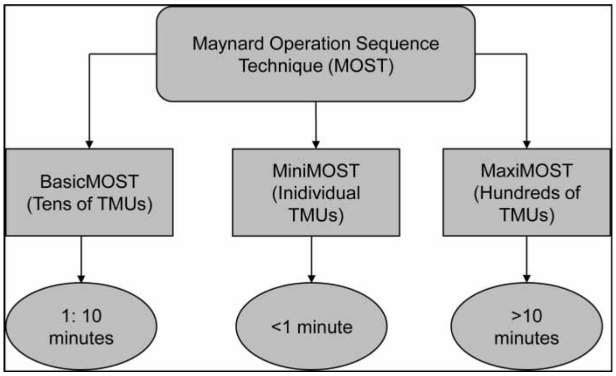

Fig. 5: Different versions of Maynard Operation Sequence Technique (MOST) [19].

Using Tab. 1 [19], the above task can be broken down into its sub-tasks as follows:

$$A1 - B0 - G1 - A1 - B0 - P3 - F10 - A0 - B0 - P0 - A0 \quad (1)$$

Thus the total time required to grasp the nut and to fasten it onto the bolt is:

$$(1 + 1 + 1 + 3 + 10)^{*}10 = 160 \text{ TMU}$$

A similar approach is used to calculate the total assembly time for the door sub-assembly using the BasicMOST technique. The total assembly time for the door sub-assembly was calculated to be 10783 TMUs which is equivalent to 6.47 minutes (1 TMU = 0.0006 minutes).

3.3. Dynamic Tree Based Optimal Precedence Sequence

As shown in the preceding sections, the liaison and precedence graph techniques can be used to automatically calculate the different assembly sequences and the times associated with the different sequences. This procedure can then be extended to calculate the optimal assembly sequence which would lead to minimum assembly time. However, calculating total assembly times for all of the feasible precedence graphs is not an efficient approach for selecting the optimal assembly sequence. Therefore, from a sequencing point of view, an optimal assembly sequence should be generated in the first attempt such that the total assembly duration for the product is minimized.

In this paper, a novel approach to generate optimal precedence sequence based on the knowledge of liaisons and using the depth first graph search algorithm [17] is presented. A dynamic tree based algorithm using the depth-first approach has been developed in this paper for generating an optimal assembly sequence which achieves the purpose of minimizing the total assembly time. In the depth first

BasicMOST System			General Move		ABGAPA		
Index*10	A: Action Distance	B: Body Motion	G: Gain Control		P: Placement		Index*10
0	<= 2 in. (5 cm)				Pickup Toss		0
1	Within Reach		GRASP	Light Object Light Objects Simo	PUT	Lay Aside Loose Fit	1
3	1-2 Steps	Sit or Stand Bend and Arise 50% occ.	GET	Light Objects Non-Simo Heavy or Bulky Blind or Obstructed Disengage Interlocked Collect	PLACE	Loose Fit Blind or Obstructed Adjustments Light Pressure Double Placement	3
6	3-4 Steps	Bend and Arise			POSITION	Care or Precision Heavy Pressure Blind or Obstructed Intermediate Moves	6
10	5-7 Steps	Sit or Stand with Adjustments					10
16	8-10 Steps	Stand and Bend Bend and Sit Climb On or Off Through Door					16

Tab. 1: BasicMOST general move data card [19].

search algorithm, all the nodes in the tree are visited by following the path joining the adjacent nodes of the current node. If there is no adjacent node available, the tree is backtracked one level up. For the automatic generation of the optimal assembly sequence, the assembly time between two subsequent operations is used as the cost function and this cost function is assigned to the edge in the tree joining the nodes corresponding to the two assembly operations.

The first step in this algorithm is to arbitrarily identify the starting component or select the starting component (SC) based on certain criteria such as size, complexity, importance to the assembly etc. Once the SC is identified, the algorithm identifies all the possible components (C_i) that can be assembled onto the SC based on the liaison information. Out of these feasible component candidates, the algorithm scans the precedence tree and identifies the component which will take the minimum time to be assembled onto the existing assembly. This component with the minimum time for assembling (calculated using MOST) will be selected as the succeeding component in the assembly. This process is repeated until the complete assembly is finished and the assembly sequence is generated. The assembly sequence created using this method will be the optimal sequence having the minimum associated assembly cost. For example, let's say that an assembly has eight components: C_1, C_2, C_3, C_4, C_5, C_6, C_7 and C_8. The liaison graphs for the eight components are shown in Fig. 6. The numbers in the bracket indicate the assembly time in minutes. The liaison connectivity for the components is as follows:

1. L_1- between C_1 and C_2
2. L_2- between C_1 and C_3
3. L_3- between C_1 and C_4
4. L_4- between C_2 and C_7
5. L_6- between C_3 and C_5
6. L_7- between C_5 and C_8

In this example, at level 1, the component C_1 is arbitrarily selected as the starting component. At the first step, the algorithm will identify all the other components that C_1 has liaison relationships with i.e. all components except C_5, C_6, C_7 and C_8. Therefore, C_5, C_6, C_7 and C_8 are neglected and the remaining components are considered as feasible candidate components. Next, using the MOST technique, assuming a crew of 2 workers, the times required to assemble each of the components with C_1 are calculated. Let's say the times for assembling the components are:

1. C_2 onto C_1: 6 min
2. C_3 onto C_1: 10 min
3. C_4 onto C_1: 11 min

Since at level 1, the assembly time for C_2 on C_1 is the least, component C_2 has to be assembled first onto C_1. Now, considering the possible options from the liaisons, using the depth first graph search algorithm, it is evident that C_6 and C_7 can be assembled onto the existing assembly i.e. C_{12}. In this paper,

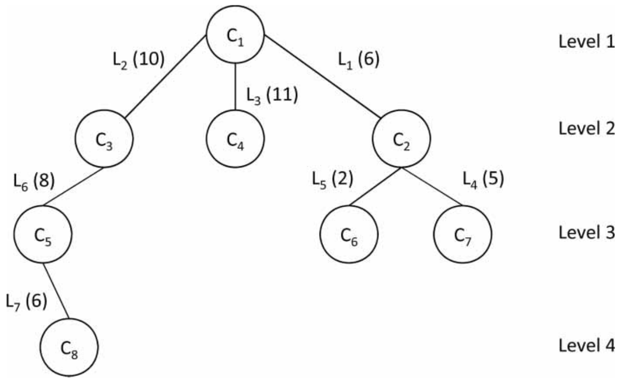

Fig. 6: Liaison diagram of an assembly with 8 components.

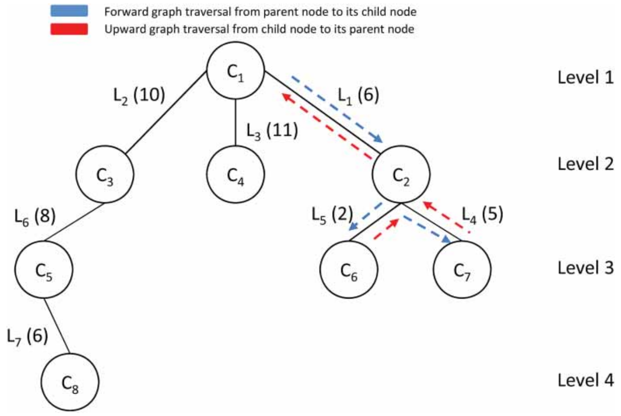

Fig. 7: Depth-First-Search for an assembly with 8 components after first step.

C_{ij} indicates the assembly obtained after assembling component C_j onto component C_i and C_{ijk} indicates the assembly obtained after assembling component C_k onto C_{ij}. Since liaison L_5 (2 min) takes less time as compared to liaison L_4 (5 min), component C_6 is assembled first onto C_{12} followed by C_7 onto C_{126}. The depth first graph search traversal is shown in Fig. 7 for the step 1. The blue arrow shows the forward graph traversal starting from the root node to the next child node while the red arrows indicates the backward travel from the child node to its parent node. If a parent node has another child, the graph traversal will begin in that direction unless all the nodes are visited for that parent node. Here, at level 1, the graph traversal sequence is C_1-C_2-C_6-C_2-C_7-C_2-C_1.

Next, the times for the assembly of remaining level 2 components onto C_{1267} are calculated as:

1. C_3 onto C_{1267}: 11 min
2. C_4 onto C_{1267}: 14 min

The assembly times for the pending components are updated in the liaison diagram after each previous assembly step as shown in Fig. 8.

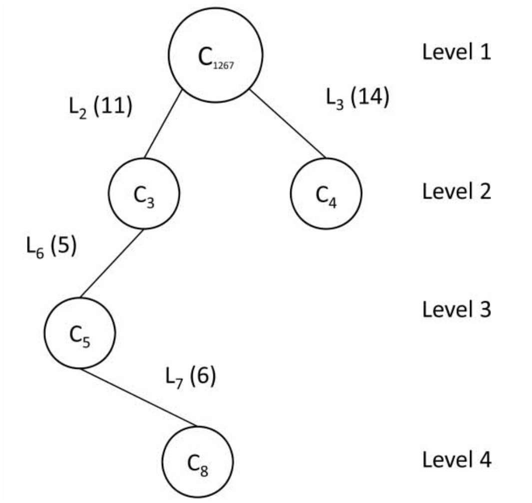

Fig. 8: Updated liaison diagram with assembly times for an assembly with 8 components.

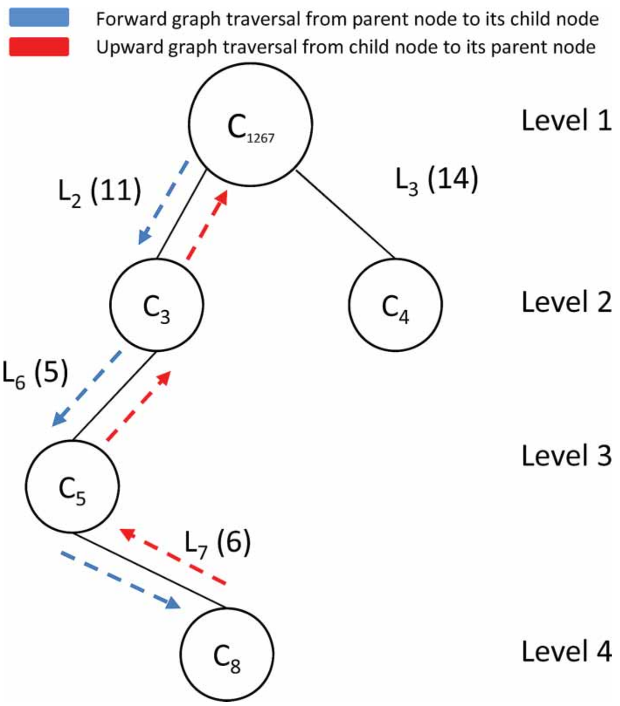

Fig. 9: Updated liaison diagram with assembly with 8 components.

It can be concluded that the next step in the assembly sequence is to assemble component C_3 onto C_{1267}. Again following the liaison structure as shown in Fig. 9, it can be seen that there exists a liaison between C_3 and C_5 and also between C_5 and C_8. The succeeding steps would therefore, be to assemble C_5 onto C_{12673} followed by assembling C_8 onto C_{126735}. The sequence for visiting the nodes at this step is C_{1267}-C_3-C_5-C_8-C_5-C_3-C_{1267}.

The last component remaining in the assembly is the component C_4 which will be assembled with the previous assembly. The idea of using liaison knowledge along with the depth first search methodology ensures the generation of assembly precedence sequence having the least possible assembly time at each level of the assembly sequence generation process. As the assembly sequence is generated by taking into account the updated assembly time at each level of the assembly, the effort of calculating the assembly time for each feasible sequence is saved.

CONCLUSIONS

An optimization model has been developed to optimize the assembly time. By incorporating the assembly time of the component while deciding the next step of the assembly sequence, the model calculates an optimal assembly sequence to finish the assembly in the least possible time. Though for explaining the overall concept only assembly time of the component is taken into consideration, the model can easily be extended with more constraints such as work crew optimization, overall labor cost, resources planning (machines etc.). The model proposed in this paper focused on a niche market that produces customized vehicles (low volume), but in future it can be extended for a mass production system.

REFERENCES

[1] Baldwin, D.F.; Abell, T.E.; Lui, M. M. and DeFazio T. L.: An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products, IEEE Jr. of Rob. and Automation, 7(1), 1991, 78–94.

[2] Bonneville, F.; Perrard, C.; Henrioud, J.M.: A genetic algorithm to generate and evaluate assembly plans, IEEE Symposium on Emerging Technologies and Factory Automation, 2, 1995, 231–239.

[3] Bourjault, A.: Contribution a une approche méthodologique de l'assemblage automatisé: elaboration automatique des séquences opértiores, Ph.D. dissertation, Université de Franche-Comte, 1984.

[4] Chen, K.; Henrioud, J.: Systematic Generation of Assembly Precedence Graphs, Proc. of the IEEE Int. Conf. on Rob. and Auto., 1994, 1476–1482.

[5] De Fazio, T.L.; Whitney, D.E.: Simplified generation of all mechanical assembly sequences, IEEE Journal of Robotics and Automation, 3, 1987, 640–658.

[6] Delchambre, A.; Wafflard, A.: An Automatic, Systematic and User Friendly Computer-Aided Planner for Robotized Assembly, Proc. of the IEEE Int. Conf. on Rob. and Auto, 1, 1991, 592–598.

[7] Delchambre, A.; Gaspart, P.: "KBAP: An Industrial Prototype of Knowledge-Based Assembly Planner", Proc. of the IEEE Int. Conf. on Rob. and Auto, 3, 1992, 2404–2409.

[8] Homen de Mello, L.S.; Sanderson, A.C.: AND/OR graph representation of assembly plans, IEEE Transactions on Robotics and Automation, 6, 1990, 188–199.

[9] Huang, Y. F.; Lee, C.S.G.: A Framework of Knowledge-Based Assembly Planning, Proc. of the IEEE Int. Conf. on Rob. and Auto, 1991, 599–604.

[10] Kanawaty, G.: Introduction to Work Study, fourth ed. International Labor Office, Geneve, 1992.

[11] Karger, D.W.; Bayha, F.H.: Engineered Work Measurement, fourth ed. Industrial Press, New York, 1987.

[12] Kaufman, S.G.; Wilson, R.H.; Jones, R.E.; Calton, T.L.; Ames, A.L.: The Archimedes 2 mechanical assembly planning system, IEEE International Conference on Robotics and Automation, 4, 1996, 3361–3368.

[13] Longo, F.; Mirabelli, G.: Effective design of an assembly line using modeling and simulation, Journal of Simulation 3, 2009, 50–60.

[14] Maynard, H.B.; Stegemerten, G.J.; Schwab, J.L.: Methods – Time Measurement. McGraw-Hill, New York, 1948.

[15] Romney, B.; Godard, C.; Goldwasser, M.; Ramkumar, G.: An efficient system for geometric assembly sequence generation and evaluation. Proceedings of the ASME International Computers in Engineering Conference, Boston, Massachusetts, 1995, 699–712.

[16] Tsao J.; Wolter J. D.: Assembly Planning with Intermediate States, Proc. of the IEEE Int. Conf. on Rob. and Auto, 1993, 71-76.

[17] Coego, J.; Mandow, L.; Pérez de la Cruz, J. L.: A Comparison of Multiobjective Depth-First Algorithms, Journal of Intelligent Manufacturing, 24, 2013, 821–829 DOI 10.1007/s10845-012-0632-y

[18] http://www.crownvic.net/drock96marquis/92-94doorpanelremoval.htm.

[19] Zandin, Kjell B.; "MOST Work Measurement Systems", 3rd Edition, Marcel Dekker Inc., NY, 2003.