Assembly Sequence Optimization

Prachi E. Ramteke
PG Student, Department of Industrial Engineering,
Shri.Ramdeobaba College of Engineering and
Management, Katol Road,Nagpur-440013, (M.S.) India

D. R. Zanwar
Associate Professor, Department of Industrial Engineering,
Shri.Ramdeobaba College of Engineering and
Management, Katol Road,Nagpur-440013, (M.S.) India

Abstract: The assembly process is one of the most time consuming and expensive manufacturing activities. Determination of a correct and stable assembly sequence is necessary for automated assembly system. The objective of the present work is to generate feasible, stable and optimal assembly sequence with minimum assembly time. Automated assembly has the advantage of greater process capability and scalability. It is faster, more efficient and precise than any conventional process. Ant Colony Optimization (ACO) method is used for generation of stable assembly sequence. This method has been applied to a Planetary Gearbox.

Keywords: Assembly Sequence, Ant Colony Optimization, Heuristic, Metaheuristic, Part Contact Level,

I. INTRODUCTION

Assembly is the process of joining separate components together to form a single final assembled unit (e.g. mechanism, device, building etc.). In many cases, the order in which these tasks are performed is an important consideration. Manufacturing and engineering design in industry require specialized knowledge and problem solving techniques. Emerging soft-computing technique can facilitate part design, process planning, scheduling, understanding and diagnosis and effective sequence determination to assemble a product. Assembly plays an important role in the design and manufacturing stages or integration of both. A wrong selection of sequence may lead to either a wrong product or halting of the process from moving towards next stage.

In production system a major role is performed by assembly process. Assembly involves several stages, such as, part handling and accessing, finding the relative position between components, selection of fixture and part placement devices, and assembly sequence determination. Sequence of assembly of a set of parts plays a key role in determining important characteristics of the tasks of assembly and of the finished assembly. It involves the identification, section and sequencing of assembly operations. The sequencing of assembly operations usually leads to the set of all feasible and reliable assembly process.

The problem of sequencing has a primary role in the development of computer based assembly systems. Different methods have been studied to approach the problem of assembly sequence; the most efficient ones are based on the application of meta-heuristic rules aimed to drastically reduce the number of sequences. A promising approach to this problem seems to be the ant colony optimization (ACO). It is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. They are inspired by the behaviour of ants in finding paths from the colony to food.

The work in this paper is based on optimising the assembly sequence of a Planetary Gearbox using ACO technique considering time. The paper is structured as follows. In Section 2, ACO is explained. The flow diagram represents the steps of ACO algorithm. Then, the basic ACO algorithm operation mode is also described In Section 3, problem is defined. The liaison diagram represents the connectivity between the mating parts of the planetary gearbox. The problem formulation is presented in Section 4. Section 5 shows the result for given problem. Finally, some concluding remarks are discussed in Section 6.

II. ANT COLONY OPTIMIZATION (ACO)

Ant Colony Optimization (ACO) was proposed by Marco Dorigo in 1992 in his PhD Thesis. The basic idea of an ant algorithm is to imitate the cooperative manner of an ant colony to solve combinatorial optimization problems within a reasonable amount of time. While building their path from nest to food source, ants can deposit and sniff a chemical substance called pheromone, which provides them with the ability to communicate with each other. An ant lays some pheromone on the ground to mark the path it follows by a trail of this substance. Ants essentially move at random, but when they encounter a pheromone trail, they decide whether or not to follow it. If they do so, they deposit their own pheromone on the path, which reinforces the trail. The probability that an ant chooses one path over others is determined by the amount of pheromone on the potential path of interest. With the continuous action of the colony, the shorter paths are more frequently visited and become more attractive for the subsequent ants. By contrast, the longer paths are less attractive because the pheromone trail will evaporate with the passing of time. Finally, the shortest way from the nest to the source of foods is found. The main characteristics of ant algorithms are positive feedback, distributed computation, and the use of a constructive greedy heuristic search.

ACO algorithm basically consists of three procedures:

1. Construct Ants Solutions
2. Update Pheromones, by which the pheromone trails are modified
3. Daemon Actions (optional), is used to implement centralized actions which cannot be performed by single ants.

1. Construct Ants solutions

Initially, each ant is put on some randomly chosen location. At each construction step, ant *k* applies a probabilistic action choice rule. In particular, the probability with which ant *k*, currently at location *i*, chooses to go to location *j* at the t^{th} iteration of the algorithm is given by Equation1

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}(t)]^{\beta}}{\sum_{l \in N_i^k}[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}(t)]^{\beta}} \quad \text{if } j \in N_i^k \qquad \text{………… (1)}$$

d_{ij} - distance from location i to location j

$\eta_{ij} = \frac{1}{d_{ij}}$ - Heuristic information, shows expectations of ants to move from location i to location j

τ_{ij} - Pheromone trial

α - determines relative influence of the pheromone trail

If $\alpha = 0$, the closest locations are more likely to be selected: this corresponds to a classical stochastic greedy algorithm (with multiple starting points since ants are initially randomly distributed on the locations).

β - determines relative influence of heuristic information

If $\beta = 0$, only pheromone amplification is at work: this method will lead to the rapid emergence of a *stagnation* situation ,that is a situation in which all the ants follow the same path and construct the same tour, which, in general, are strongly suboptimal.

N_i^k - Feasible neighbourhood of ant k,i.e. the set of locations the ant has not visited yet.

2. Pheromone update

After all ants have constructed their tours, the pheromone trails are updated. This is done by first lowering the pheromone strength on all arcs by a constant factor and then allowing each ant to add pheromone on the arcs it has visited, is given by Equation 2.

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \qquad \text{………… (2)}$$

Where, $0 < \rho < 1$ is the pheromone trail evaporation. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails and it enables the algorithm to “forget" previously done bad decisions. If an arc is not chosen by the ants, its associated pheromone strength decreases exponentially.

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{if arc } (i,j) \text{ is used by ant} \\ 0 & \text{otherwise} \end{cases} \qquad \text{....……….. (3)}$$

Where, $L^k(t)$ is the length of the k^{th} ant's tour

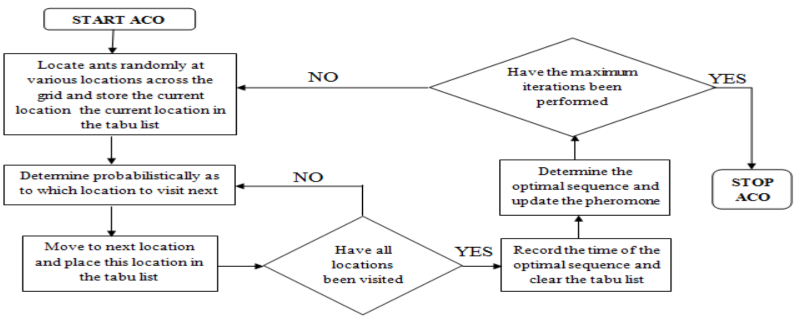
Q is a constant.

3. Daemon Actions (optional)

It is used to implement centralized actions which cannot be performed by single ants.

As a practical example, the daemon can observe the path found by each ant in the colony and select one or a few ants (e.g., those that built the best solutions in the algorithm iteration) which are then allowed to deposit additional pheromone on the components/connections they used.

Flow diagram of ACO algorithm



ACO Algorithm

The Pseudo-code for the ACO algorithm is as follows:
/*Initialization*/
For every edge (i, j) do
$\tau_{ij}(0) = \tau_0$
End For
For k=1 to m do
Place any ant at randomly chosen city
End For
Let T^k be the shortest tour found from beginning and L^k its length
/*Main loop*/
For t=1 to t_{max} do
For k=1 to m do
Build tour T^k (t) by applying (n-1) times the following step
Choose the next city with probability

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha . [\eta_{ij}(t)]^\beta}{\sum l \in N_i^k [\tau_{ij}(t)]^\alpha . [\eta_{ij}(t)]^\beta} \qquad \text{if } j \in N_i^k$$

Where i is the current city
End For
For k=1 to m do
Compute the Tour T^k and the length L^k produced by ant k
End For
If an improved tour is found
update T^k and L^k
End If
For every edge (i, j) do
Update pheromone trail by applying the rule:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^k(t)$$

where

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{if arc } (i,j) \text{ is used by ant} \\ 0 & \text{otherwise} \end{cases}$$

End For
For every edge (i, j) do
$\tau_{ij}(t+1) = \tau_{ij}(t)$
End For
Print the shortest Tour and length.
Stop

III. PROBLEM DEFINATION

In order to work on the proposed method, product (planetary gearbox) is considered as the case study. The planetary gearbox has 34 parts. Several parts overlapped in the CAD data, including 16 screws which were modelled larger than their corresponding holes. Figure (1) represents an exploded view of a planetary gearbox. Table (1) represents the Spare Parts List of a Planetary Gear Box and Table (2) represents Connectivity of the parts with each other. It shows the work element description.

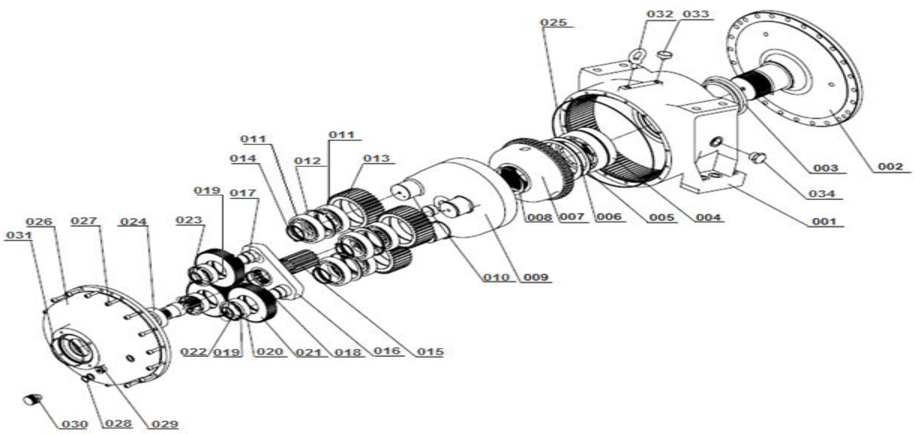


Fig. 1 An Exploded view of Planetary Gearbox

TABLE 1
SPARE PARTS LIST

Items	*Qty/GBox*	*Name*	*Items*	*Qty/GBox*	*Name*
01	1	Main Housing	18	1	Snap Ring
02	1	Output Flange	19	6	Snap Ring
03	1	Flange Main Seal	20	3	Radial Ball Bearing
04	1	Main Output Flange Bearing	21	3	Snap Ring
05	1	Supporting Ring	22	3	Planetary Gear
06	1	Snap Ring	23	1	Sun Gear
07	1	Convex Gear	24	1	Radial Ball Bearing
08	1	Snap Ring	25	1	O – Ring
09	1	Planetary Carrier	26	1	Rear Cover
10	1	Lead Pivot	27	16	Cheese Head Screw
11	6	Cylindrical Roller Bearing	28	1	O – Ring(Copper)
12	3	Snap Ring	29	1	Magnet
13	3	Planetary Gear	30	1	Oil Draining Safety Valve
14	3	Snap Ring	31	1	O – Ring
15	1	Sun Gear	32	1	Eye Bolt
16	1	Planetary Carrier	33	1	Breather Bolt
17	3	Planetary Carrier pivot	34	1	Oil Gauge

TABLE 2
WORK ELEMENT DESCRIPTION

Elements	*Work Element Description*	*Time (sec)*
1	Load 1	3
2	2 in 1	10
3	3 in 2	4
4	4 in 2	4
5	5 in 2	3
6	6 in 2	3
7	7 over 2	5
8	8 in 7	3
9	9 over 7	7
10	10 in 9	3

11	11 in 9	3 x 3s = 9
12	12 in 9	3 x 3s = 9
13	11 in 9	3 x 3s = 9
14	13 over 11 and 12	3 x 3s = 9
15	14 in 9	3 x 2s = 6
16	15 in 9	6
17	16 in 15	8
18	18 in 16	3
19	19 in 17	3 x 3s = 9
20	20 over 17	3 x 3s = 9

Elements	*Work Element Description*	*Time (sec)*
18	18 in 16	3
19	19 in 17	3 x 3s = 9
20	20 over 17	3 x 3s = 9
21	21 over 17	3 x 3s =
22	19 in 17	3 x 3s = 9
23	22 in 17	3 x 2s = 6
24	23 in 16	4
25	24 over 23	3
26	25 in 1	3
27	26 over 1	7
28	27 screwed in 1	16 x 5s = 90
29	28 in 26	3
90	29 in 26	3
31	30 in 26	4
32	31 in 26	3
33	32 in 1	3
34	33 in 1	3
35	34 in 1	3

1. Liaison Diagram of Planetary Gearbox

Liaison means communication between two or more groups, or co-operation or working together. Liaison diagram is similar to ER diagram. It builds relationship between entities and checks all possibilities for precedence and considers the feasible ones.

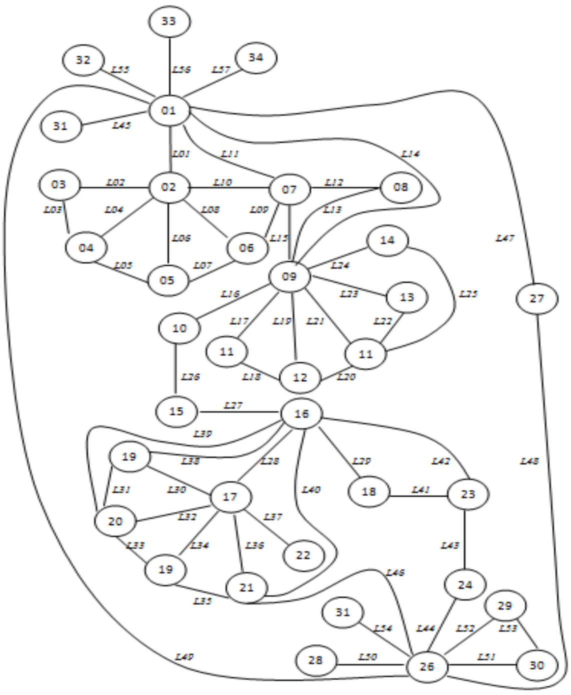


Fig. 2 Liaison diagram

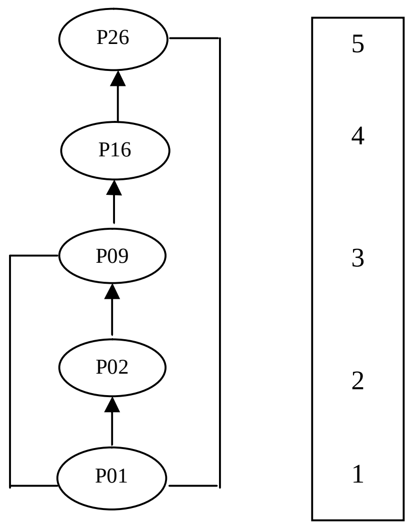


Fig. 3 Part contact Level Graph

2. Part Contact Level Graph

The part contact level graph is used to study relationships between individuals, groups, organizations, or even entire societies. The term is used to describe a structure determined by such interactions. The ties through which any given unit connects represent the convergence of the various contacts of that unit. In above case of a planetary gear box assembly, the parts P01, P02, P03, P04 and P05 are the sub assembled parts which are in contact with each other. It gives an idea about the assembly of the product.

By utilizing the part assembly precedence constraints thus obtained, the feasible assembly sequence of the base parts is obtained as,

$$P01 \rightarrow (P02 \rightarrow P09) \rightarrow P16 \rightarrow P26$$

The part assembly precedence constraint, therefore, is a key factor for inferring the assembly sequence. This inference procedure becomes more systematic and simple, when compared to that of conventional geometrical reasoning or query/answer methods.

The parts P01, P02, P09, P16 and P26 are the subassemblies of planetary gearbox. The complete assembly of a planetary gearbox is to be done by joining these sub assembled parts with each other with optimal sequence and in minimum time.

Time taken to assemble the subassemblies P01, P02, P09, P16 and P26:

P01 (task1) = P25 + P32 + P33 + P34 = 3 + 3 + 3 + 3 = 12 sec

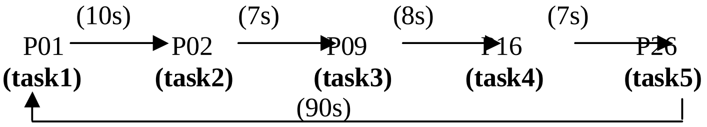
P02 (task2) = P03 + P04 + P05 + P06 + P07 + P08 = 4 + 4 + 3 + 3 + 5 + 3 = 22 sec

P09 (task3) = P10 + P11 + P12 + P11 + P13 + P14 + P15 = 3 + 9 + 9 + 9 + 9 + 6 + 4 = 49 sec

P16 (task4) = P18 + P19 + P20 + P21 + P19 + P22 + P23 + P24 = 3 + 9 + 9 + 9 + 9 + 6 + 4 + 3 = 52sec

P26 (task5) = P28 + P29 + P30 + P31 = 3 + 3 + 4 + 3 = 13sec

Time taken to assemble these subassemblies:



Time = 10 + 7 + 8 + 7 + 90 = 122sec

To solve the ACO matrix the parameters which are considered are:

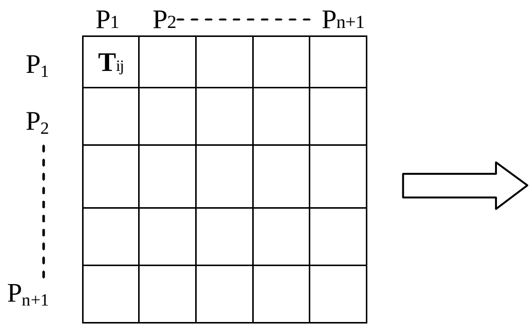
n- number of task = 5

task time = (1→2 = 10s), (2→3 = 7s), (3→4 = 8s), (4→5 = 7s), (5→1 = 90s)

task dependency = 1 > 2, 1 > 3, 1 > 5, 2 > 3, 3 > 4, 4 > 5

IV. PROBLEM FORMULATION

In order to assess the performance of ACO, sample problems are considered formulated and evaluated. The problem formulation consists of time matrix generation. The time matrix formulates the problem in 2D matrix. Firstly, inputs are given in the form of number of task, task time and task dependency. Then the location of the task is decided on the grid on both X-axis and Y-axis. The processing time matrix is used to represent the processing time expected for every task on every resource. The time matrix generated is then given as the input to ACO. The matrix shown below is the time matrix.



	1	2	3	4	5
1	0	10	10	10	10
2	7	0	7	7	7
3	8	8	0	8	8
4	7	7	7	0	7
5	90	90	90	90	0

The above time matrix gives the optimised time for the given sequence. This process has to be repeated till the optimised time with the optimal sequence is obtained. Number of iterations is performed for better results.

V. RESULT

The results obtained by using one of the effective metaheuristic optimization tools, viz. ant colony optimization (ACO) to solve the assembly sequence generation are shown below.

The base assembly sequence, as mentioned above using part contact level of planetary gearbox is found out to be

$$P01 \rightarrow (P02 \rightarrow P09) \rightarrow P16 \rightarrow P26$$

The complete sequence of the assembly as obtained by the process is presented below,

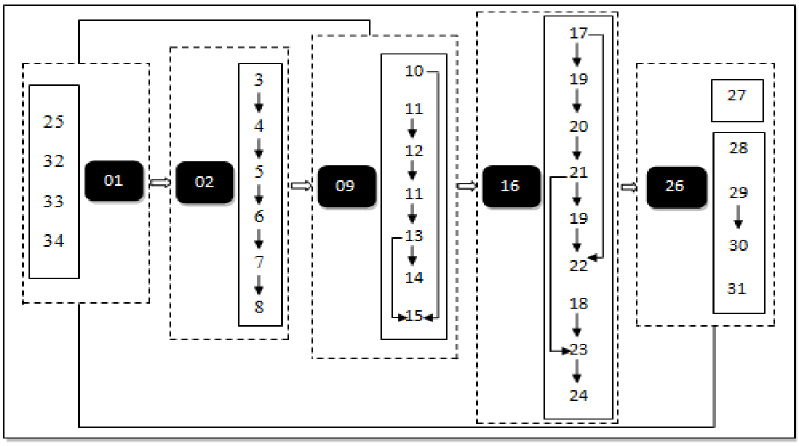


Fig. 4 Complete Assembly Sequence

Results of Assembly Sequence Order

TABLE 3
ASSEMBLY SEQUENCE ORDER

Seq. No.	***Sequence Order***				
01	1	2	3	4	5
02	1	2	4	5	3
03	1	3	2	4	5
03	1	4	2	3	5
04	1	5	4	3	2
05	1	5	3	2	4
06	2	1	3	5	4
07	2	1	3	4	5
08	2	3	1	4	5
09	3	1	2	4	5
10	3	2	1	4	5
11	4	1	2	3	5
12	4	3	5	1	2
13	4	3	5	1	2
14	4	3	2	1	5
15	4	5	1	2	3
16	4	5	2	1	3
17	4	5	3	1	2
18	4	5	3	2	1
19	5	1	2	3	4
20	5	1	3	2	4
21	5	1	4	3	2
22	5	2	1	4	3
23	5	3	4	1	2
24	5	4	1	2	3
25	5	4	2	1	3

Feasible Assembly Sequence

The feasible assembly sequences from the above assembly sequence order are taken considering Figure 4

TABLE 4
FEASIBLE ASSEMBLY SEQUENCE LIST

Seq. No.	*Optimised Assembly Sequence*				
01	1	2	3	4	5
02	1	3	2	4	5
03	1	5	4	3	2
04	2	1	3	4	5
05	2	3	1	4	5
06	3	1	2	4	5
07	3	2	1	4	5
08	4	3	5	1	2
09	4	5	3	1	2
10	4	5	3	2	1
11	5	1	4	3	2

Optimised Assembly Sequence

Best Feasible Sequence with optimised time from the above feasible sequence is,

TABLE 5
OPTIMISED ASSEMBLY SEQUENCE

Seq. No.	*Optimised Assembly Sequence*					*Opt. Time(s)*	*Time Saved(s)*	*Time Saved (%)*
01	2	1	3	4	5	109	13	10.65

VI. CONCLUSION

For a wide variety of manufacturing industries, assembly accounts for a significant percentage of the total manufacturing cost of a product which is directly dependent on time required to manufacture the product. Until recently, much effort has been devoted to designing and producing individual parts, while very little effort has been directed towards optimizing product assembly. Hence, it is likely that an improvement in this area could yield a great reduction in both manufacturing cost and time. Following the present trend it can be forecast that in the future an increasing number of products will be designed for automatic robotic assembly.

Although extensive research works have been carried out, several avenues for future research still remain open. Several methods for the generation of assembly sequences are discussed in course of the present work. However, the core work concentrates on important and simple methods for selecting the appropriate method for automated assembly system. Nevertheless, sufficient research outcome may be realized in some of the following areas.

1) Automating the sequence generation would be much useful in the field of product assembly.

2) The reduction in assembly time may be achieved by designing products in such a way that they are easy to handle and assemble. In order to assemble the newly designing products, simpler and less expensive robots are required.

REFERENCE

[1] De Fazio T. and Whitney D., “Simplified generation of all mechanical assembly sequences”, *IEEE J. Robotics and Automation*, 1987.

[2] Baldwin D., Abell T., MaxLui M., De Fazio T. and Whitney D., “An integrated computer aid for generating and evaluating assembly sequences for mechanical products”, *IEEE Trans. on Robotics and Automation* 7, No. 1, 1991.

[3] Chan P., “Automatic assembly sequence generation by pattern matching”, *IEEE Transactions on Systems, Man and Cybernetics’*, 1991.

[4] Homan de Mello L. S. and Sanderson A. C., “Representations of mechanical assembly sequences”, *IEEE Transactions on Robotics and Automation*, 1991.

[5] Young-Hyun Han., Sebti Foufou., and Shaw C. Feng., “A Model for Capturing Product Assembly Information” Journal of Computing and Information Science in Engineering MARCH 2006, Vol. 6 / 11.

[6] S.Sharma, B.B.Biswal, P.Das and B.B.Choudhury “Generation of Optimized Robotic Assembly Sequence using Ant Colony Optimization”, Proc. of IEEE CASE, Washington DC, USA, 2008.

[7] Dorigo M et al., “The ant system: optimization by a colony of cooperating agents”, *IEEE Trans Syst Man Cyber Net*, 1996.

[8] Dorigo, M., “Ant colony system: A cooperative learning approach to the travelling salesman problem”, *IEEE Transactions on Evolutionary Computation. IEEE*, 1(1), 1997

[9] Lee S., “Assembly planning based on subassembly extraction” *Trans. Robotics and Automation*, Special Issue on Manufacturing Systems: Design, Planning, Scheduling and Real-Time Control, 1990, pp.1-32.

[10] Lee S. and Shin Y. G., “Assembly planning based on subassembly extraction” *Proc. 1990 IEEE International Conference on Robotics and Automation*, 1990.

[11] Lee S. H, “Disassembly planning by subassembly extraction,” *Proc. 3rd ORSA/TIMS Conf. Flex. Manuf. Sys.*, 1989.

[12] Cho and Cho H., “Inference on robotic assembly precedence constraints using part contact level graph” *Robotica*, Vol. 11, 1993.

