Optimal Robotic Assembly Sequence generation using Particle Swarm Optimization

M V A Raju. Bahubalendruni
Dept. of Industrial Design, National Institute of Technology- Rourkela, Odisha-769008, India
Email: first. bahubalindruni@gmail.com

B.B. Biswal and B.B.V.L Deepak
Dept. of Industrial Design, National Institute of Technology- Rourkela, Odisha-769008, India
Email: bbbiswal@nitrkl.ac.in and bbv@nitrkl.ac.in

***Abstract* - The optimal feasible robotic assembly sequence leads to efficient manufacturing process by minimizing the assembly cost. Assembly cost is based on the energy required to assemble the components through collision free path and robot directional changes during the assembly operations. So, the determination of a feasible assembly sequence with minimum assembly cost is vital concern for manufacturing industries. Through obtaining optimal assembly sequences taking user inputs (assembly connection matrix, precedence relations, etc.,) is less complicate, the correctness of methodology depends on the skill of the engineer who supply these inputs. The present research aims to explore PSO based methodology to determine cost effective optimal robotic assembly sequence through CAD product. The integration of PSO with CAD environment ensures the correctness and completeness of the methodology. The methods to interface with the CAD data to extract liaison data, to test for liaison predicate and feasibility predicate is presented and analyzed briefly with an example. In this methodology, each component of the assembled product is considered as the particle (bird) and mutation operation is performed to generate a new assembly sequence for each iteration. To generate optimal assembly sequence, a fitness function is generated, which is based on the energy function and robot directional changes associated with assembly sequence. The sequence which is having the best fitness value is treated as the optimal robotic assembly sequence.**

Index Terms—: robotic assembly sequence, particle swarm optimization, optimal assembly sequence.

I. INTRODUCTION

The assembly sequence of a product plays very important role in manufacturing because it consists of 10-30% of the manufacturing time, minor change in assembly motion and change in assembly direction can influence the cost of manufacturing great extent. To reduce such assembly cost, much research effort has been made on assembly sequence generation.

Assembly sequence directly influences the productivity of the process, product quality, and the cost of production. The robotic assembly process is faster, efficient and precise than any conventional process. The ratio between cost and performance of assembly has gradually increased with respect to the other phases of the manufacturing process and in recent years, because of this fact researcher's interest is growing in this field. An important aspect of this developing process is represented by the need to automatically generate the assembly plan by identifying the optimum sequence of operations with respect to its cost and correctness. Products with large number of parts have several alternative feasible sequences among which optimal assembly sequence is generated.

Baldwin et al.[1] developed simplified method which can find the optimal solutions, but have a problem of the search space explosion for an increased number of parts. Hong and Cho[2-4] proposed neural-network based computational approaches, which have been reported to overcome the problem of the search space explosion. However, the methods have a problem of frequent generation of no optimal sequences, since the network energy often reaches to a local minimum. Cho and Cho [5] developed a method using directional part contact level graphs which contains the information on directional connections for each pair of mating parts. Lee [6] proposed disassembly method. In this method, an assembly sequence was determined by the reverse order of disassembly sequence expressed in a list of parts each of which is sequentially chosen to have minimum cost of disassembly. These are some of the classical approaches for solving assembly sequencing plan. Besides the above mentioned techniques, researchers have also concentrated on artificial intelligence techniques for

solving the same problem but with less mathematical complexity. Wang et al.[7] proposed ant algorithm by using the disassembly operations of the parts in assembly sequence planning. Smith and Lui[8] had used the most common evolutionary algorithm, Genetic Algorithm (GA) to generate robot assembly sequences. This methodology generates the optimal assembly sequence by minimizing the assembly cost while satisfying the assembly constraints. Schutte et al.[9] implemented PSO algorithm for the biomechanical optimization and conclude that PSO algorithm is easier to be fulfilled than GA algorithm. Wang[10] proposed a variation of PSO in solving the same. Zhang et al.[11] used a discrete particle swarm optimization (DPSO) algorithm to solve the multiple destination routing problems. Chen[12] proposed an adaptive particle swarm optimization approach to solve the problem of minimizing the printed circuit board assembly time simultaneously with optimization of assignment problems for a pick-and-place Machine. Liao et al.[13] resolve the complex job-shop scheduling problem using an improved PSO algorithm in which local heuristic information is introduced. Shen et al[14] proposed an improved fuzzy discrete particle swarm optimization method and applied it to traveling salesman problem. Bahubalendruni et al.[15-16] proposed computer aided methods to extract the assembly connections, and efficient method to test feasibility predicate from CAD environment.

In most of the research done in past three decades, the input to the optimization algorithm was given manually in terms of liaison matrix/assembly connection matrix and precedence relations between the components. Hence the results are dependent on the skill of the person who is supplying these inputs. However, as analyzed by Bourjault and Defazio[17,18]. Obtaining the precedence relations for an assembly involved in generation lot of questions and answering those by a skillful person. Hence the optimization of robotic assembly sequences is not fully automated without any user intervention. Thus, the current research is completely dedicated to develop a methodology to integrate the optimization algorithm with CAD environment to extract all the necessary information to generate cost effective optimal robotic assembly sequences with correctness and completeness without any user intervention. And also an efficient method of feasibility checking is depicted and integrated with the algorithm.

II. Overview of Methodology

The proposed methodology interface with the 3D CAD environment to get the liaison matrix, to test the feasibility of assembling operation and to compute the energy involved in assembling operations. In this section a brief note on liaison connectivity test, feasibility test and energy computation. The flowchart depicted in Figure 1 briefs the methodology of obtaining the optimal feasible robotic assembly sequences.

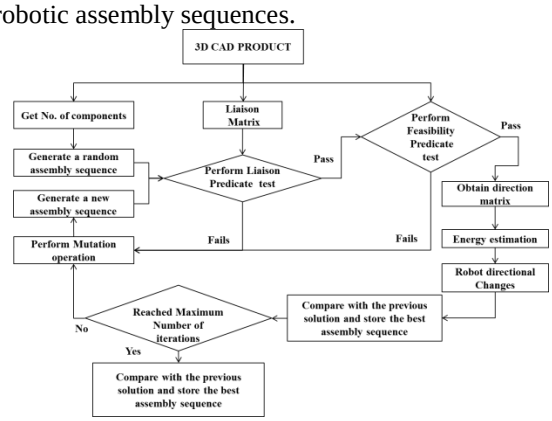

Figure 1. Method of robotic assembly sequence generation

A. *Liaison establishment test*

Liaison diagram is a concept of representing the liaisons between pairs of components to describe the significant relationships between the parts of an assembly, this method is initially proposed by Bourjault[17] and later popularized by De Fazio[18]. A liaison is a defined connections established between the components.

Matrix representation of liaisons is proposed by Dini[19] using binary codes 1,0. A nxn matrix is required to represent the liaisons connections for a product assembled by "n" components. The diagonal elements of this matrix will consist null values, and the row of matrix represents the liaisons between one component with the other components in the assembly. The column of matrix represents the components connected by liaison relationships. The sub-matrices of nxn matrix represent the local liaison relationships in subassemblies.

Algorithm to extract the liaison matrix from 3D solid models is presented below

open an assembly in CAD environment
obtain the number of parts in the Assembly "say ***n*** *number of parts"*
create a null matrix of "nxn"
compute the conflicts between all components
obtain total number of conflicts "m"
for each conflict 1 to m
    define the conflict type by conflict value

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 jth part

replace the null value with "1" for the [i][j] and [j][i] elements of null matrix

end If

end for

export the liaison matrix data

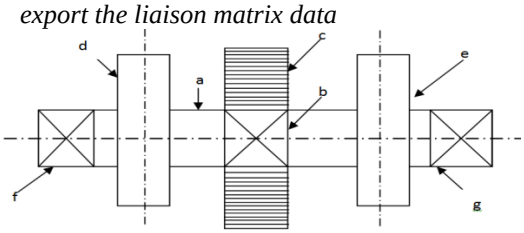

Figure 2. Representation of 7 part Gear assembly [a-shaft; b-bearing; c-gear, d-Arm, and e-Arm, f-nut, g-nut]

A gear assembly composed of 7-parts shown in Figure 2 is considered to illustrate the methodology, for which liaison matrix is generated from CAD environment through the presented code is mentioned below.

	A	B	C	D	E	F	G
A	0	1	0	1	1	1	1
B	1	0	1	0	0	0	0
C	0	1	0	0	0	0	0
D	1	0	0	0	0	0	0
E	1	0	0	0	0	0	0
F	1	0	0	0	0	0	0
G	1	0	0	0	0	0	0

Joining a component to a part/subassembly create an assembly can be possible when the component has at least one connection with any part of the subassembly.

For the assembly sequence [A-D-E-F-C-G-B] the part-C cannot be assembled to the product ADEF, due to the reason that part C has no significant established connection with any of the parts in the ADEF subassembly. So that each assembly sequence must be qualify the liaison predicate test. The liaison predicate test identifies, is there exist a liaison to assemble the component to the existed product/part.

Methodology to test the liaison establishment is presented below

for i= part 2 to n-1.

temp ← 0

for j= part 1 to i-1.

temp ← temp + liaisonmatrix[i][j]

end for

if temp=0

go to mutation operation to generate new assembly sequence

end if

end for

There exist n! robotic assembly sequences, approximately 40-70% of assembly sequences are eliminated at this phase, and the qualified sequences will only be passed to the next stage for the feasibility test.

B. Feasibility predicate test

A part can bring into contact with its mating parts through any collision free path, then the part is said to be feasible to assemble. The feasibility predicate can be tested based on the assumption that "if a part can be disassembled from the product without any collision, then the part can also be assembled". When the feasibility predicate is true for each part in the assembly sequence, the assembly sequence will be considered for energy estimation.

For the assembly sequence A -E-D-B-C-F-G, feasibility testing is represented in Table.1.

TABLE.1 FEASIBILITY TEST RESULTS FOR AN ASSEMBLY SEQUENCE.

Assembled Product	**Part to be removed**	**Possibility**
A -E-D-B-C-F-G	G	Yes
A -E-D-B-C-F	F	Yes
A -E-D-B-C	C	**No**

The possibility of removing each part must be checked along five directions (X+,X-, Y+, Y- and Z+) assuming that the assembly is place on a base plate and the disassembling procedure cannot be possible in Z- direction. Since testing feasibility in all directions is too tedious, it is efficient to check from low distance direction to high distance direction. The distance to be moved by the component to assembly can be obtained using the bounding boxes for each component. Representation of bounding boxes for assembled product and the component to be removed is illustrated in Figure 3.

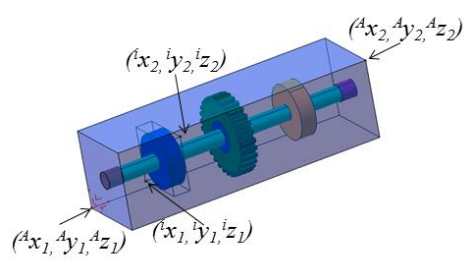

Figure 3. Representation of bounding boxes at component and assembly level

TABLE.2 DISTANCE TO BE MOVED BY THE COMPONENTS TO ASSEMBLE/DISASSEMBLE

	Part "*i*"				
Disassemble directions	X+	X-	Y+	Y-	Z+
Assemble directions	X-	X+	Y-	Y+	Z-
Distance to be moved	${}^{A}x_2 - {}^{i}x_1$	${}^{i}x_2 - {}^{A}x_1$	${}^{A}y_2 - {}^{i}y_1$	${}^{i}y_2 - {}^{A}y_1$	${}^{A}z_2 - {}^{i}z_1$

The distances to be moved by each component to disassemble from assembled product in all possible five directions are listed in table.2. The directions must be arranged in ascending order based on the distance to be moved and checking for feasibility in the same order minimizes the time and efforts. The bounding box corner coordinates for the gear assembly shown in Figure 3 are listed in table.3 and the distances to be moved by part D from the product is listed in table.4.

TABLE.3 COMPONENT AND ASSEMBLY LEVEL BOUNDING BOX CORNERS.

	x_1	y_1	z_1	x_2	y_2	z_2
Assembly	0	0	0	340	100	100
A	0	40	40	340	60	60
B	160	30	30	180	70	70
C	160	0	0	180	100	100
D	70	15	15	90	85	85
E	250	15	15	270	85	85
F	0	40	40	10	60	60
G	330	40	40	340	60	60

TABLE.4 DISTANCE TO BE MOVED BY THE COMPONENTS TO ASSEMBLE/DISASSEMBLE

	Part 4				
Disassemble directions	X+	X-	Y+	Y-	Z+
Assemble directions	X-	X+	Y-	Y+	Z-
Distance to be moved	${}^{A}x_2 - {}^{4}x_1$	${}^{4}x_2 - {}^{A}x_1$	${}^{A}y_2 - {}^{4}y_1$	${}^{4}y_2 - {}^{A}y_1$	${}^{A}z_2 - {}^{4}z_1$
	270	**90**	**85**	**85**	**85**

From the table, it is efficient to check the feasibility to disassemble the part "D" in the following directions "Z+,Y+ , Y-, X-,X+". Though the distance to be moved are same in "Z-" direction and "Y±" directions, priority will be given to the 'Z' direction due to gravity force.

Methodology to test the feasibility predicate is presented below

for i= part n to 2.

for j= 1 to 5 (directions arranged in ascending order for i^{th} part)

for k=0 to distance along j direction

move the part to a distance "k" along j^{th} direction

perform contact analysis

if there exist interference then

if j=5 then

assembly sequence is not feasible

go to mutation operation to generate new assembly sequence

end if

change the direction (go to next j value)

end if

if k= distance along j direction then

compute energy for the operation

go to the next part

end if

end for

end for

end for

There will not be any feasibility check for the last part, since it can be disassemble in all the possible directions, the lowest distance direction will be given to it. The feasible assembly sequences will be transferred to the next phase for energy computation for the assembly process.

C. Energy estimation

The qualified assembly sequence in liaison establishment test and feasibility predicate test is considered for energy estimation. The energy to disassembly product can be considered as the product of density, volume and the distance moved by the part. For the assembly sequence [A-F-D-B-C-E-G] total energy calculated by using the expression (1) using the values listed in table.5.

$$\sum_{i=1}^{n} \rho_i V_i d_i \qquad (1)$$

TABLE.5 COMPONENT VOLUME, DENSITY AND DISTANCES

Parts	A	F	D	B	C	E	G
Disassemble directions	Z+	X-	X+	X+	X+	X+	X+
Distance to be moved (d) X 10^{-2}	60	10	270	180	180	90	10
Volume (V) X 10^{-6}	106.81	2.36	70.69	18.85	114.05	70.69	2.36
Density (ρ)	7860	7860	7860	7860	7860	7860	7860

As the direction differs from one part to the next part, robot has to change its directions accordingly.

The energy consumption due to the robot directional changes also influences the overall assembly lead time and energy. The energy consumption to change the robot direction is mainly dependent on the specifications of the robot, however minimal directional changes in the assembly sequence results in less energy consumption. Hence the current objective is to find out an assembly sequence with less energy consumption having minimal changes in the assembly directions. The below is computer aid to obtain the number of directional changes for an assembly sequence.

Methodology to obtain the number of robot directional changes

```
count1=0
for i= 1 to n-1
  dif= dir(i+1) - dir(i)
 if dif=0
 count1=count1+1
end for
count=n-count
```

The number of directional changes for a specified assembly sequence can be obtained using the above mentioned code through the direction matrix associated with the assembly sequence. An assembly sequence with minimum number of directional changes with same energy level will be considered as optimal assembly sequence.

III. Implementation of Particle swarm optimization

PSO is a population based methodology, which was inspired by social behaviour of bird flocking or fish schooling. The population considered in PSO is called swarm and its individuals are known as particles. So a swarm in PSO can be defined as a set $S = \{P_1, P_2, P_3, \dots, P_n\}$. Where $P_1, P_2, P_3, \dots, P_n$ is 'n' number of particles in the swarm. These particles are assumed to move within the search space. While the particles are moving, their new positions can be updated with a proper position shift called velocity. Let us consider the positions of 'n' particles are: $\{x_1, x_2, x_3, \dots, x_n\}$ and their velocities are: $\{v_1, v_2, v_3, \dots, v_n\}$. The new velocity of each particle is obtained from the communicated information of particles among the swarm. It can be done in terms of memory i.e. each particle stores its best position, it has ever visited during its search. The best position decided by each particle is called position best and is indicated by X_{pbest}. So there are 'n' number of position best values for 'n' particles in the swarm.

Now the particles in the swarm are mutually communicated their experience and they will approximate to one global best position, ever visited by all particles as shown in Figure 4.

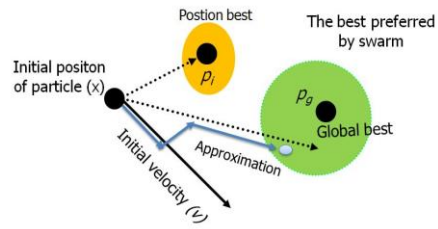

Figure 4. Basic structure of PSO for global best approximation

Selection of global best position can be done by calculating the fitness values of each particle in the swarm. The particle which is having the best fitness can be treated as the global best position and is represented by X_{gbest}. The determination of X_{gbest} indicates the completion of one PSO-iteration. This process will be continued until maximum number of iterations has occurred or robot has reached its target.

In this paper PSO with mutation operation is used to generate an assembly sequence. Later, optimal assembly sequence is obtained by calculating its fitness value with following steps:

Step1: consider each part (a,b,c…) as the each individual in the swarm. And initialize position and velocity for each individual randomly for *1* to *n* (number of parts) as illustrated in Table 6.

TABLE 6. INITIAL POSITION AND VELOCITY OF EACH INDIVIDUAL (PART)

Individual Part name	a	b	c	. . .	p
Position x(i)	1	2	3	. . .	n
Velocity v(i)	1	2	3	. . .	n

Step2: Apply mutation operation for two parts by keeping one fixed part with respect to all other parts. For example, a product is constructed with three individual parts say a,b and c and allocate its position values randomly a to 1, b to 2 and c to 3 as illustrated in Table.7. Then apply mutation operation to the primary sequence 'a-b-c' such that 'a' (fixed part) with respect to 'b' and then 'c'. So the generated sequence in the second and third iterations will be b-a-c and c-b-a respectively.

TABLE.7. POSITION VALUE OF EACH PART DURING MUTATION OPERATION

Position x(i)			
Random sequence	a	b	c
2nd iterative sequence	b	a	c
3rd iterative sequence	c	b	a

Step3: updating position and velocity of each individual (part).

To update the position and velocity of each particle, a new parameter is introduced here named as 'position shift' as follows:

Position shift = position value (second part) - position value (first part) (2)

Position update: updated position of the particle is according to the equation as follows.

$x_i(t+1) = x_i(t)$ + position shift (3)

Finding Xgbest & Xpbest:

Xgbest can be obtained after applying mutation operations to one fixed part with respect to all other parts. During the mutation operation with respect to a fixed part, which sequence is giving the optimal fitness value followed by assembly constraints is treated as Xgbest. In table.8, the initial position of 'a' is 1 but after mutation operation the position of 'a' is 2. Let us consider b-a-c is having optimal fitness value, Xgbest for this iteration will be {2, 1, 3}. Xpbest of each part is nothing but the new position of the corresponding part.

TABLE.8. REPRESENTATION OF XPBEST & XGBEST

Parts	X (i)	X (i+1)	Position shift	Xp best	Xg best
a	1	2	1	2	Best sequence either of tth or $(t+1)^{th}$ iteration according fitness value.
b	2	1	-1	1	
c	3	3	0	3	
					For 2nd iteration '2'

The PSO cycles will be processed as follows:

1st cycle:

Consider initial position of each particle in swarm as its position best.

Initialize X_{gbest} as the generated assembly sequence.

Apply mutation between last two allocated parts

Calculate position shift of particle using Eq. (2).

Obtain updated positions of particle using Eq. (3).

2nd cycle:

Load updated positions of each particle from first cycle.

Consider new positions of particles as their position bests.

Find X_{gbest} by calculating fitness of obtained sequence using Eq.(1).

Calculate position shift of particle using Eq. (2).

Calculate new positions of particle using Eq. (3).

3rd cycle:

Load updated positions of each particle from second cycle.

Consider new positions from 2nd cycle as their position bests.

Find X_{gbest} by calculating fitness of obtained sequence using Eq.(1).

Calculate position shift of particle using Eq. (2).

Calculate new positions of particle using Eq. (3).

... and so on

Step4: once the robotic assembly sequence is generated in each iteration, its feasibility is to be checked.

If the generated sequence is feasible, the next step is to find out its fitness value using eq.(1). Later the fitness of the updated sequence is to be compared with previous sequence fitness. If the updated sequence is giving the best fitness value then PSO iterations will be continued with new sequence, otherwise cycles will be continued with the earlier sequence.

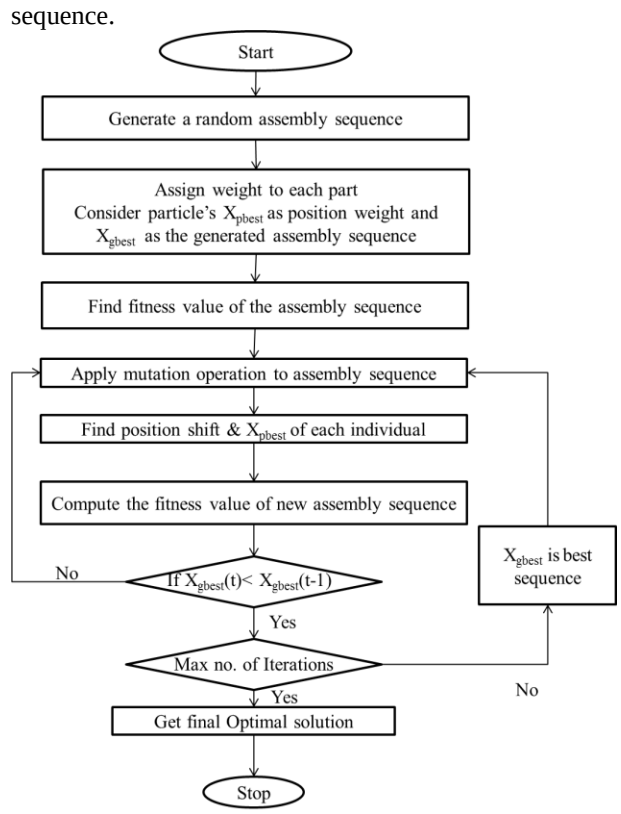

Figure 5. PSO implementation for optimal assembly sequence generation

IV. IMPLEMENTATION AND RESULTS

When the PSO based methodology applied on the 7 part gear assembly shown in Figure 2, the resulted assembly sequences, the respected disassemble direction array along with the energies and the robot directional changes are listed on iteration basis. Till the 46th iteration, the methodology is unable to generate at least one feasible assembly sequence. A feasible assembly sequence is detected at 47th iteration and assembly sequence with same energy consumption with minimum directional changes has been replaced at 69th iteration.

An assembly sequence with low energy level is found at 178th iteration and is continued till iteration 209. The assembly sequence with same energy level and minimum number of directional changes till the maximum number of iterations reached. Table.9 lists the outcomes of the methodology at different iteration levels. Graphical representation of convergence is represented in Figure 6.

TABLE.9 ASSEMBLY SEQUENCE AND ENERGY LEVELS

Iteration No.	Assembly Sequence	Disassemble direction array	Energy(j)	No of Robot directional changes
1	Non feasible sequences			
47	D-A-B-C-E-F-G	Z+||X+,X+,X+,X+||X-||X+	56.3076	3
69	D-A-B-C-E-G-F	Z+||X+,X+,X+,X+,X+||X-	56.3076	2
118	B-A-C-D-E-F-G	Z+||X+,X+||X-||X+||X-||X+	54.9745	5
167	A-F-D-B-C-E-G	Z+||X-||X+,X+,X+,X+,X+	43.2651	2
178	A-B-D-C-E-F-G	Z+||X+||X-||X+,X+||X-||X+	33.4038	5
210	A-B-C-D-F-E-G	Z+||X-,X-,X-,X-||X+,X+	33.4038	2

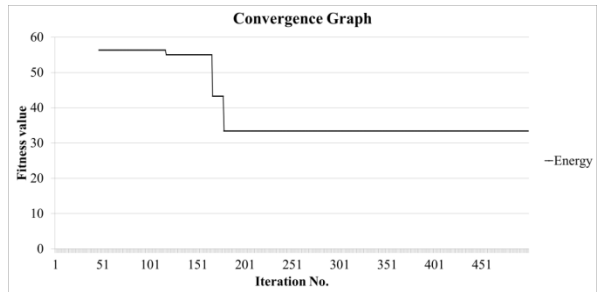

Figure 6 Iteration based convergence curve

V. Conclusion

The activities of generation of liaison matrix from CAD product, testing for liaison predicate, feasibility predicate from CAD product using an efficient method is well described. PSO based methodology has been developed to generate the feasible and optimal robotic assembly sequence with minimum assembly cost. A clear explanation has been given in order to find out the assembly sequence from the possible number solutions. During the implementation, each part of the assembled sequence is considered as a particle. For the generated assembly sequence, after applying mutation operation in each iteration, the sequence is checked for liaison predicate and feasibility predicate. For all feasible sequences, fitness value is calculated and comparison of fitness values has been done between consecutive generated sequences. Then the mutation operation is applied to find our optimal fitness valued sequence.

REFERENCES

1. D. Baldwin, T. Abell, M. Lui, T. De Fazio and D. Whitney, "An integrated computer aid for generating and evaluating assembly sequences for mechanical products", *IEEE Trans. Robot. Automat.*, vol. 7, no. 1, pp. 78-94, 1991.
2. D. Hong and H. Cho, "Optimization of robotic assembly sequences using neural network", *Intelligent Robots and Systems" 93. Proceedings of the 1993 IEEE/RSJ International Conference on*. 1993; 1: pp232-239.
3. D. Hong and H. Cho, "A neural-network-based computational scheme for generating optimized robotic assembly sequences", *Engineering Applications of Artificial Intelligence*, vol. 8, no. 2, pp. 129-145, 1995.
4. D. Hong and H. Cho, "A genetic-algorithm-based approach to the generation of robotic

assembly sequences", *Control Engineering Practice*, vol. 7, no. 2, pp. 151-159, 1999.
5. D. Cho and H. Cho, "Inference on robotic assembly precedence constraints using a part contact level graph", Robotica, vol. 11, no. 02, pp. 173-183, 1993.
6. S. Lee, "Disassembly planning by subassembly extraction", *Proceedings of the third ORSA/TIMS Conference on flexible manufacturing systems*. 1989. pp.383-388.
7. J. Wang, J. Liu and Y. Zhong, "A novel ant colony algorithm for assembly sequence planning", Int J Adv Manuf Technol, vol. 25, no. 11-12, pp. 1137-1143, 2005.
8. S. Smith and Y. Liu, "The application of multi-level genetic algorithms in assembly planning", *J Ind Technol*, vol. 17, no. 4, pp.1-9, 2001.
9. J. Schutte, B. Koh, J. Reinbolt, R. Haftka, A. George and B. Fregly, "Evaluation of a Particle Swarm Algorithm For Biomechanical Optimization", Journal of Biomechanical Engineering, vol. 127, no. 3, pp. 465-474, 2005.
10. Y. Wang and J. Liu, "Chaotic particle swarm optimization for assembly sequence planning", *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, pp. 212-222, 2010.
11. Z. Zhang and C. Huang, "A Discrete Particle Swarm Optimization Approach to Optimize the Assembly Sequence of Mechanical Product", *AMR*, vol. 490-495, pp. 203-207, 2012.
12. Y. Chen and C. Lin, "A particle swarm optimization approach to optimize component placement in printed circuit board assembly", *Int J Adv Manuf Technol*, vol. 35, no. 5-6, pp. 610-620, 2007.
13. C. Liao, Chao-Tang Tseng and P. Luarn, "A discrete version of particle swarm optimization for flowshop scheduling problems", *Computers & Operations Research*, vol. 34, no. 10, pp. 3099-3111, 2007.
14. B. Shen , M. Yao and W. Yi, "Heuristic information based improved fuzzy discrete PSO method for solving TSP", *Trends in Artificial Intelligence*. pp. 859-863.
15. MVAR., Bahubalendruni and BB. Biswal, "Computer Aid for Automatic Liaisons Extraction from Cad Based Robotic Assembly" *Proceedings of the 8th International Conference on Intelligent Systems and Control (ISCO)*, pp.42-45,2014.
16. MVAR, Bahubalendruni and BB. Biswal, "An Algorithm to Test Feasibility Predicate for Robotic Assemblies", *Trends in Mechanical Engineering & Technology, vol.* 4, no. 2, pp: 11-16, 2014.
17. A. Bourjault, "Contribution a une Approche Methodologique de L"Assemblage Automatise: 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,1984.
18. T. De Fazio and D. Whitney, "Simplified generation of all mechanical assembly sequences", *IEEE J. Robot. Automat*., vol. 3, no. 6, pp. 640-658, 1987.
19. G. Dini, F. Failli, B. Lazzerini and F. Marcelloni, "Generation of Optimized Assembly Sequences Using Genetic Algorithms", *CIRP Annals - Manufacturing Technology*, vol. 48, no. 1, pp. 17-20, 1999.

Dr. Bibhuti Bhushan Biswal graduated in Mechanical Enginnering from UCE, Burla, India in 1985. Subsequently he completed his M.Tech, and Ph.D. from Jadavpur University, Kolkata. He joined the faculty of Mechanical Engineering at UCE Burla from 1986 and continued till 2004 and then joined National Institute of Technology, Rourkela as Professor and currently he is the Professor and Head of Department of Industrial Design. He has been actively involved in various research projects and published more than 100 papers at National and International levels. His areas of research interest include industrial robotics, FMS, Computer integrated manufacturing, automation, and maintenance engineering. He was a visiting Professor at MSTU, Moscow and a visiting scientist at GIST, South Korea.

Mr. M V A Raju Bahubalendruni post graduated in Machine Design specialization from Andhra University, India in 2007 and Joined in HCL Technologies as an aerostructural design engineer and worked in several aircraft redesign programs till 2012. He has expertise in composite aircraft structures, computer aided design automation and design and optimization of mechanical structures. Currently he is perusing Ph.D. in National Institute of Technology, Rourkela. He has published 2 patents and more than 10 research articles in reputed journals and conferences. His areas of research interest contain CAD Automation, FDM, Design optimization and Composite structures.

Dr. B B V L Deepak post graduated in Machine Design specialization from National Institute of Technology, Rourkela , India in 2011 and Joined as

senior research fellow in National Institute of Technology, Rourkela 2011. He completed his Ph.D. degree from National Institute of Technology, Rourkela in 2015. He has expertise in Mechatronics, Mobile robots, Industrial automation, Optimization engineering He has published more than 25 research articles in reputed journals and conferences.