ROBOTIC ASSEMBLY SEQUENCE GENERATION USING IMPROVED FRUIT FLY ALGORITHM BY APPLYING GROSS AND FINE MOTION AS FITNESS EQUATION

Gunji Bala Murali*1, Sridharan2, BBVL Deepak3, Rajashekara Reddy Muthra4, BB Biswal5, MVA Raju Bahubalendruni6

1, 2, 4 Department of Design and Automation, Vellore Institute of Technology, Vellore, Tamilnadu, India-632014.
3, 5 Department of Industrial Design, National Institute of Technology, Rourkela, India-769008.
6 Department of Mechanical Engineering, National Institute of Technology, Pudhucherry, India- 609609.

*balamurali.g@vit.ac.in, 2sridharan.k@vit.ac.in, 3bbvl@nitrkl.ac.in, 4rajasekhara.reddy@vit.ac.in,
5bbbiswal@nitrkl.ac.in, 6bahubalindruni@gmail.com.

Abstract. Robotic assembly plays a major role in manufacturing industries to reduce the cost of overall product. As the assembly is one of the major manufacturing processes, which involves much information like contact information between the parts, tools needed during assembly, part orientation information during assembly and many more. Most of the researchers considered grippers changes, assembly time, directional changes, part orientation and energy of the part as fitness function. In this research work authors made attempt to consider gross and fine motion as fitness equation to generate optimal assembly sequences with Improved Fruit Fly Algorithm (IFFA). The proposed algorithm is implemented on many industrial products to obtain the optimal assembly sequences.

Keywords: Assembly Sequence planning, Gross, Fine motion, Fruit Fly Algorithm

1. Introduction
Robotic assembly is now replacing the traditional assembly process in the manufacturing due to high flexibility during assembly, more accurate and less to time assembly. Due to these many advantages, modern manufacturing industries replacing the traditional assembly process to robotic assembly. But robotic assembly is not as simple as traditional assembly because it requires lot of assembly information like type of tools used to assemble, type of gripper used to hold the object, contact relation between the parts and many more [1]. Most of the researchers are concentrated on contact information, gripper information, and feasibility information to obtain the optimal assembly sequences. At the initial stages of ASP problem solving, authors like Wilson and Latombe [2] applied mathematical models to generate solution for the ASP problem. As these methods only generate feasible solutions, they are absolute by raise in demand from the customers in terms of cost and quality.

Meanwhile, advancements in Artificial Intelligence (AI) changed the world to shift the interest towards AI for solving the engineering problems. Keeping the advantages in view regarding to the AI techniques, researchers applied Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm, Simulated Annealing (SA) and so on to solve ASP problem [3-5].
Along with the general AI algorithms that are existed, the authors like Murali, Gunji Bala, et al. [6,7] applied Improved Cuckoo Search (ICS) and Crab Shell Search (CSS) algorithm to generate optimal assembly sequences. Out of these, CSS is a newly developed algorithm based on how crab will search the shell at the shore to survive from the foreign bodies. As the individual algorithms are restricted to certain limits like struck at local solution and highly redundant in nature, Hybrid Algorithms (HA) are developed by the researchers to solve ASP problems. HA utilizes the advantages of the algorithms that are combining to solve ASP problem [8]. Most of the methods/techniques used to solve ASP problem uses the either mathematical approach or connector base approach to extract the assembly predicates. But Gunjia, Bala Murali, et al. applied Computer Aided Design (CAD) based method to extract the assembly predicates [9]. Along with the traditional methods Murali, G. Bala, et al. proposed assembly subset detection method using stability graph to generate optimal assembly sequences. In this method, authors try to implement the subset detection method at each stage to eliminate the higher level subsets by evaluating the fitness at each stage. By this, time of execution and search space is reduced compared to the existing methods [10]. As of now the discussed methods/algorithms to solve ASP problems uses directional changes or gripper changes or energy to carry the part or assembly tool changes as fitness function to generate optimal assembly sequences. In this paper, the authors made an attempt to generate optimal assembly sequences by considering gross and fine motion as fitness function.

To generate optimal assembly sequences using gross and fine motion as fitness equation, the authors applied Fruit Fly (FF) algorithm by improving search phenomenon of the fly. The paper is organised as follows: section-1 deals with introduction and literature of the ASP problem. Section-2 explains about the improvement made for the fruit fly algorithm and how it is implemented to solve ASP problem. Section-3 deals with the results obtained from the proposed methodology. Section-4 explains about conclusion made from the proposed method in the paper.

2. Improved Fruit Fly Algorithm

In this section, FF algorithm is used to generate optimal assembly sequences for the considered seven part gear assembly by improving the search phenomenon of the fruit fly. FF algorithm is basically developed by the inspiration of natural behaviour of the fruit fly in search of ripened or fermenting fruits during summer season. Fruit flies are having good senses and perceptions than the other species. The individual fruit fly uses its sensitive vision to fly in the direction of the target by sampling the different scents that are present in the surroundings. As the individual fly identifies the best position that is nearer to the target, it directs the flocking location of the flies closer to the food. Through the iterative evolution, the fly swarm comes closer and closer to the target until it reaches to the fruits.

By using this phenomenon, fruit fly algorithm was developed by Pan, Wen-Tsao for the engineering problem for financial distress model. Later this algorithm was extended to many fields. In general fruit fly algorithm, initialization of the fruit fly is considered by assigning the random position in plane consisting of food as shown in the Figure 1. Due to this, the selected random position may not be in the direction of food, in that case the solution occurs is of redundant.
In the improved Fruit Fly algorithm instead of generating random positions to find local best, a range has been considered to generate positions of the Fruit Fly to find local best. The range is defined by a fly angle of Fruit Fly from initialization position of Fruit Fly. The range is decided by considering the food position region as the 2D plane with the coordinate system.

To avoid this phenomenon, improvement is made in the search position to find the local best solution for the fruit fly algorithm as shown in Figure 2.

In the Figure 2, if it is considered as 2D plane, then the range is given as 0-90° for the search position in the algorithm. By which location of food will compulsory in that region where the solution is not redundant.

The detailed proposed methodology is explained in Figure 3 with the flow chart to generate optimal assembly sequences. In the applied algorithm, gross and fine motions are considered as fitness equations to generate optimal assembly sequences. The generation of fitness equation in terms of gross and fine motion is shown in Eq (1).
ff = \sum_{i=1}^{n} GM_i + FM_i \quad (1)

Where ff= fitness function
n= Part number
GM=Gross Motion
FM=Fine Motion

2.1. Gross and Fine Motion

Gross and fine motions are two motions that robot arm will possess during assembling the parts. Gross motion is the motion carried by the robot arm with high speeds during assembly. These motions are existed till the parts come closer just before parts are in contact. Fine motion is the motion that will comes into existence once the part come contact with the other part during the assembly. These motions are very fine motions to have precision assembly and firm contact between the parts. The details about the fine and gross motion are explained with the screw jack diagram shown in the Figure 4.

Assumption made in this paper: As the robotic assembly consists of robotic work cell, where the individual parts have been kept at different places and should bring to the robotic base is where robot has been fixed by the robotic arm. During bringing of the parts to the base, the robot arm has to follow gross motion concept to reduce the time of assembly. But in this paper, this has not been considered as we are not dealing with the practical assembly.
Figure 4. Represents the gross and fine motion for the screw jack assembly

3. Results and discussion
In this section, a seven part gear assembly shown in Figure 5 is considered to generate optimal assembly sequences using gross and fine motions as fitness values.

Figure 5. Represents the gear assembly

Part-1: Shaft; Part-2: Bearing; Part-3: Gear; Parts-4&5: Rotating discs; Parts-6&7: Holders.

Based on the contact relation graph, the algorithm generates feasible solution depending on the population size. After generation of the feasible solution, fitness is evaluated for the sequences to generate optimal assembly sequences. The contact relation graph is shown in Figure 6, by which the information about the contact between the parts is assigned to the algorithm in the form matrix.
Figure 6. Contact relation graph

Contact relation matrix
This matrix summarizes the contact information that has to feed as input to the algorithm to generate feasible sequences. The detailed information about the matrix is as follows:

Contact relation matrix

\[
\begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
1 & -1 & 0 & -1 & 0 & 0 & 0 \\
2 & 0 & -1 & 0 & -1 & -1 & -1 \\
3 & -1 & 0 & -1 & -1 & -1 & -1 \\
4 & 0 & -1 & -1 & -1 & -1 & -1 \\
5 & 0 & -1 & -1 & -1 & -1 & -1 \\
6 & 0 & -1 & -1 & -1 & -1 & -1 \\
7 & 0 & -1 & -1 & -1 & -1 & -1
\end{bmatrix}
\]

Generally the matrix is represented by ‘0’ s and -1 s to provide the information about the contact relation between the parts. In the matrix ‘0’ represents the contact existed between the parts and ‘-1’ represents the no contact between the parts.

As the assembly is not having the gross motion because as there is no part dimensions which are out of the shaft dimensions so, fine motion is only existed. The fine motion distance has been extracted from the CATIA V5 R17 Digital mock up tool bar. The fine motion distances are shown in table-1.

Table-1: Represents the fine motion distances

| Part name     | Fine/Gross motion distance (mm)          | Direction of assembly |
|---------------|------------------------------------------|-----------------------|
| Bearing       | 90 (80 fine and 10 gross)               | +Z or -Z              |
| Gear          | 90 (80 fine and 10 gross)               | +Z or -Z              |
| Rotating disc | 53 (43 fine and 10 gross)               | +Z                    |
| Rotating disc | 53 (43 fine and 10 gross)               | -Z                    |
| Holder        | 10 fine                                 | +Z                    |
| Holder        | 10 fine                                 | -Z                    |

By providing the contact relation information and fine motion information to the algorithm, the algorithm is made to run for 200 iterations to generate optimal assembly sequences. As the gross and
fine motion are in mm, the total distance moved for any assembly sequence will be same in order to avoid this problem, the distance moved will be converted as time to show the difference in the fitness.

**Assumption:** The time taken to move 10mm distance in gross motion will be consider as 5sec. and the time taken to cover the 10mm distance in fine motion will be 10 sec.

A graph has been plotted between number of iterations and time taken to assembly by following the assembly sequence is shown in Figure 7. From the graph after ‘65’ iterations, the solution is converged and the fitness value is of ‘456 Sec.’. The lists of assembly sequences that are generated from the algorithm, which will take 456sec to assemble the parts, are listed in table-2.

| S. No. | Assembly Sequence | Time taken (sec) |
|--------|-------------------|-----------------|
| 1      | 1-2-3-4-5-7-6     | 456             |
| 2      | 1-2-3-5-4-6-7     | 456             |
| 3      | 1-4-2-3-5-7-6     | 456             |
| 4      | 1-5-2-3-4-6-7     | 456             |

![Figure 7. Represents the graph between fitness value and number of iterations](image)

**4. Conclusion**

In this paper, an attempt is made to generate optimal assembly sequences by considering gross and fine motions as fitness function. A gear assembly with seven parts has been considered to generate the sequences using improved fruit fly algorithm. The results obtained are very impressive in terms of time taken to assembly the parts if a particular assembly sequence is followed. As a future scope, robotic work cell dimensions will be taken into consideration for more practical approach of implementing the generated sequences to robot to assembly the parts.

**References**

[1]. Deepak, B. B. V. L., et al. "Assembly sequence planning using soft computing methods: a review," Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering **233.3** (2019): 653-683.

[2]. Wilson, Randall H., and Jean-Claude Latombe. "Geometric reasoning about mechanical assembly." Artificial Intelligence **71.2** (1994): 371-396.

[3]. FAILLI, FRANCO, and Gino Dini. "Ant colony systems in assembly planning: a new approach to sequence detection and optimization." 2nd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering-ICME 2000 **1.0** (2000):227-232.
[4]. Murali, G. Bala, et al. "Integrated design for assembly approach using ant colony optimization algorithm for optimal assembly sequence planning." *Computational intelligence in data mining*. *Springer, Singapore 0.0 (2019)*: 249-259.

[5]. Murali, G. Bala, et al. "Optimal assembly sequence planning towards design for assembly using simulated annealing technique." *International Conference on Research into Design*. *Springer, Singapore 0.0 (2017)*: 397-407.

[6]. Murali, Gunji Bala, et al. "Robotic Optimal Assembly Sequence Using Improved Cuckoo Search Algorithm." *Procedia computer science* **133** (2018): 323-330.

[7]. Murali, Gunji Bala, et al. "A New Crab Shell Search Algorithm for Optimal Assembly Sequence Generation." 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON). *IEEE, 0.0 (2019)*; 108-114.

[8]. Abdel-Baset, Mohamed, and Ibrahim Hezam. "Cuckoo search and genetic algorithm hybrid schemes for optimization problems." *Appl Math* **10.3** (2016): 1185-1192.

[9]. Gunjia, Bala Murali, et al. "CAD-based automatic clash analysis for robotic assembly." *International Journal of Mathematical, Engineering and Management Sciences* **4.2** (2019): 432-441.

[10]. Murali, G. Bala, et al. "Optimal robotic assembly sequence planning using stability graph through stable assembly subset identification." *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* **233.15** (2019): 5410-5430.

[11]. Pan, Wen-Tsao. "A new fruit fly optimization algorithm: taking the financial distress model as an example." *Knowledge-Based Systems* **26** (2012): 69-74.