HYBRIDIZED CUCKOO-BAT ALGORITHM FOR OPTIMAL ASSEMBLY SEQUENCE PLANNING

Balamurali Gunji[1, a], B.B.V.L.Deepak[1,b], Amruta Rout[1,c], Golak Bihari Mohanta[1,d]
B.B.Biswal[1 e]

[1] Industrial Design Department, NIT Rourlela, Orissa, Pin-769008, India.
[a] bmgunji@gmail.com, [b] bbv@nitrkl.ac.in, [c] 516id6003@nitrkl.ac.in,
[d] 516id1001@nitrkl.ac.in, [e] bbbiswal@nitrkl.ac.in

Abstract. Assembly Sequence Planning (ASP) problem is one of the NP-hard combinatorial problems in manufacturing, where generating a feasible sequence from the set of finite possible solutions is a difficult process. As the ASP problem is the discrete optimization problem, it takes a major part of the time in the assembly process. Many researchers have implemented different algorithms to get optimal assembly sequences for the given assembly. Initially, mathematical models have been developed to solve ASP problems, which are very poor in performance. Later on, soft computing techniques have been developed to solve ASP problems, which are very effective in achieving the optimal assembly sequences. But these soft computing techniques consume more time during execution to get optimal assembly sequence. Sometimes these algorithms fall in local optima during execution. Keeping the above things in mind in this paper, a new algorithm namely Hybrid Cuckoo-BAT Algorithm (HCBA) is implemented to obtain the optimal assembly sequences. The proposed algorithm is compared with two different assemblies (gear assembly and wall rack assembly) with the algorithms like Genetic Algorithm (GA), Ant Colony Optimization (ACO), Grey Wolf Optimization (GWO), Advanced Immune System (AIS) and Hybrid Ant Wolf Algorithm (HAWA). The results of the different algorithms are compared in terms of CPU time and fitness values with the proposed algorithm. The results show that the proposed algorithm performs better than the compared algorithms.

Keywords: Assembly Sequence Planning Problem, Objective Constraints, Input Constraints, Soft Computing Techniques.

1. Introduction

Assembly is the process of joining the parts in an order one after the other to form assembly. To join the part sin an order, one requires assembly sequence, which will give the information about the parts that are to be joined to form assembly. As the assembly sequence planning problem is the discrete optimization problem, achieving the optimal assembly sequence is difficult process. To achieve the optimal assembly sequence, the generated sequence from any method has to undergo two criteria. One is feasibility criteria, to satisfy this criterion the assembly sequence has to check with the input constraints (liaison data, stability data). If the assembly sequence satisfies the feasibility criterion then, it has to check for the second criterion to increase the quality of the sequence by evaluating through objective constraints. Many researchers used different methodologies/algorithms to obtain the optimum assembly sequence. At the initial stage of developments in the ASP problems, researchers are used mathematical models like liaison graph/ liaison tree to check the feasibility of the sequences, which is time consuming process. Later on, computer aided methods has been developed to extract the input constraints automatically [1-3]. In order to have good quality sequence, initially researchers are followed the mathematical algorithms like cut-set methods, AND/OR questions to obtain the optimal assembly sequence. But these are very tedious and time consuming [4, 5]. Later on researcher's developed the computer aided techniques to obtain the optimal assembly sequences. Generally these methods are classified in to two types: one is graph search algorithm and second one is Artificial Intelligence (AI) based algorithms. Even though these methods are successful to achieve optimal assembly sequences, but sometimes these methods fall in the local optima during execution.

To overcome this, researchers are attracted towards the hybrid algorithms. As the hybrid algorithms are the combination of two or more algorithms desired features top obtain the optimal assembly sequence [6-8].

In the current research, a new HCBA has been developed to obtain the optimal assembly sequence. The developed algorithm is compared with the different well known algorithms like GA, ACO, AIS, GWO and HAGA.

2. Literature review

Solving the ASP problem started in late 1980's by Ayoub and Doty [9]. Later on it De Mello and Sanderson developed AND /OR graph to obtain the optimal assembly sequence [5].The researchers like Chakrabarty and Wolter [10] developed a hierarchy of assembly structure to reduce the complexity of the problem. Later, the researchers like Xiaoming and Pingan [11] developed object oriented method to obtain the optimal assembly sequence.

The above discussed methods consume lot of execution time as well as search space also. To avoid this problem researchers are motivated towards the soft computing techniques. Till now many soft computing techniques are applied by different researchers to obtain the optimal assembly sequences. Out of those, GA is used by most of the researchers because of its simplicity in implementing. Initially GA is implemented by Wong and Leu [12] to obtain the optimal assembly sequence. In this he implemented adaptive GA by continuously varying the genetic operators. Later the researchers like Boizneville et al.[13] , Dini et al. [14], Hong and Cho [15] and Smith and Liu [16] developed the GA to solve optimal ASP problem. Out of those Smith and Liu uses multi-level genetic algorithm is used, in which the sequence obtain from the level -1 will be given ass input to the level-2 by which in feasible solutions will reduces and quality of the solution will increase. Apart of GA the next mostly used algorithm by the researchers is ACO algorithm. This algorithm is initially implemented by Failli and Dini [17] to solve assembly sequence planning problem. Later on it was developed by Wang et al. [18], McGovern and Gupta [19] and Wang et al. [20]. Out of them Wang uses dis-assembly feasibility graph to obtain the optimal assembly sequence. Apart of these algorithms, many recently developed algorithms like Advanced Immune Strategy proposed by Bahubalendruni [21], Grey Wolf Optimization algorithm proposed by Mirjalili, S et.al. [22] and many more have been implemented to solve asp problem.

The rest of the paper is arranged as follows: section-3, deals with the proposed algorithm, section-4 deals with the results and comparisons of the proposed algorithm and section-5 deals with the conclusion of the research paper.

3. Proposed Algorithm

In this section a new hybrid algorithm is proposed to obtain the optimum assembly sequence. In this, cuckoo search and bat algorithms are combined to form a hybrid algorithm, to achieve optimal assembly sequence. In the proposed algorithm two fitness functions have been considered for two separate assemblies to evaluate the quality of the assembly sequence.

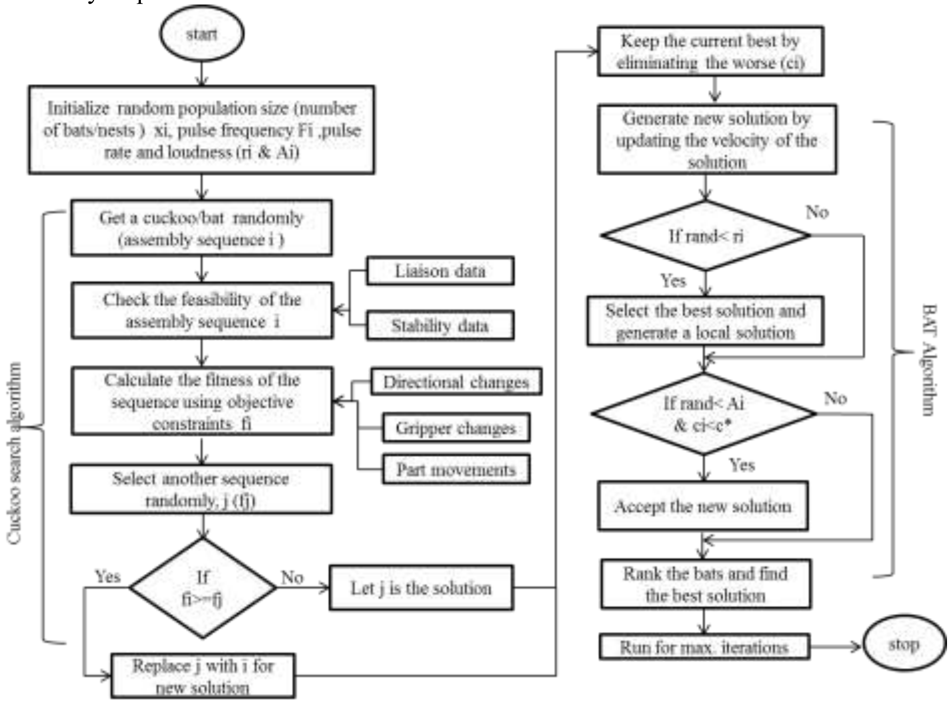

Figure -1: Flow chart of hybrid cuckoo –bat algorithm

The detailed flow chart of the developed algorithm is shown in the figure-1. The fitness functions for both the assemblies are as follows:

For the first assembly three objective constraints are considered to evaluate the fitness of the sequence. In this, Directional Changes$(D.C)$, Gripper Changes $(G.C)$ and Part Movement $(P.M)$ are considered for developing the fitness function.

Fitness Function $F.F = \sum_{i=1}^{n-1} \frac{w}{3} * (D.C_i) + \frac{w}{3} * (G.C_i) + \frac{w}{3} * (P.M_i)$ (1)

Where w is the weight function of the different constraints, which depends on the industry requirement.
n- is the number of parts in the assembly

$$If\ w = \begin{cases} 1 & \text{Three objective constraints are having equal priority} \\ 3/2 & \text{Any two objective constraints are having equal priory and other is '0'} \\ 3 & \text{Only one objective constrains is having full priority and rest are '0'}^{s} \end{cases}$$

The formulation of the fitness function by giving equal priority to the three objective constraints is as follows:

$$F.F = \sum_{i=1}^{n-1} 0.33 * (D.C_i) + 0.33 * (G.C_i) + 0.33 * (P.M_i) \qquad (2)$$

To compare the proposed algorithm, the second assembly considered is wall rack. For the wall rack assembly to evaluate the quality of the sequence, the fitness function is formulated by considering the directional changes and gripper changes as objective constraints. The formulation of the equation is as follows:

$$F.F = \sum_{i=1}^{n-1} w * (D.C_i) + (1-w) * (G.C_i) \quad (3)$$

Let us considered both objective constraints are having the equal priority, then

$$w = 0.5 => \quad F.F = \sum_{i=1}^{n-1} 0.5 * (D.C_i) + 0.5 * (G.C_i) \quad (4)$$

4. Results and Comparisons

This section deals with the extracted results of the two industrial products namely gear assembly and wall rack assembly from the proposed algorithm. In this, the results are compared with the different algorithms like Genetic Algorithm (GA), Ant Colony Optimization (ACO), Grey Wolf Optimizer (GWO), Hybrid Ant –Wolf Algorithm (HAWA) and Advanced Immune Based Strategy. In this two assemblies are been considered to compare with all algorithms.
In this two cases have been considered separately for two assemblies. In the first case, gear assembly shown in the figure-2 is considered for evaluating the quality of the sequence by the proposed methodology. The results of the proposed algorithm are compared with the advanced immune algorithm [21].

Case-1: Gear Assembly

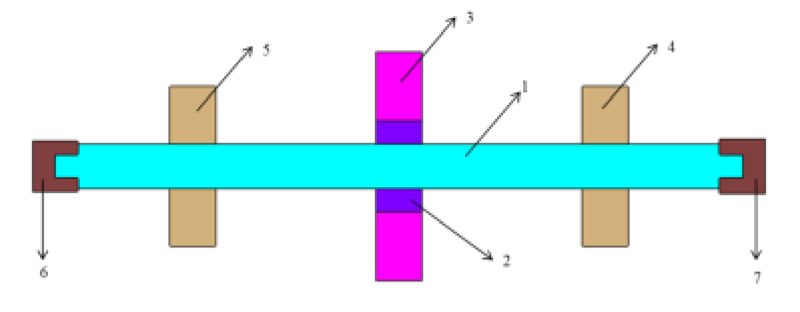

Figure-2: Gear assembly

Contact Data: This data provides the information about the contact of the parts in the assembly. In the below matrix '0' represents no contact between the parts and '1' represents the contact between the parts.

Liaison data

	1	2	3	4	5	6	7
1	0	1	0	1	1	1	1
2	1	0	1	0	0	0	0
3	0	1	0	0	0	0	0
4	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0
6	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0

Stability Data: This matrix provides the information about the stability of the parts in the assembly. In the below matrix '0' represents the no stability, '1' represents the partial stability and '2' represents the permanent stability respectively.

Stability data

	1	2	3	4	5	6	7
1	0	1	0	1	1	2	2
2	1	0	1	0	0	0	0
3	0	1	0	0	0	0	0
4	1	0	0	0	0	0	0
5	1	0	0	0	0	0	0
6	2	0	0	0	0	0	0
7	2	0	0	0	0	0	0

Geometrical feasibility matrices: This data provides the information about the feasibility direction of part for the assembly. These matrices are total six in six principle axes. In this '0' represents feasible in that direction and '1' represents not feasible in that direction.

+x -x +y

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	0	0	1	1	1	1
4	0	1	1	0	1	1	1
5	0	1	1	1	0	1	1
6	0	1	1	1	1	0	1
7	0	1	1	1	1	1	0

	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	0	1	1	1	1
3	0	0	0	1	1	1	1
4	0	1	1	0	1	1	1
5	0	1	1	1	0	1	1
6	0	1	1	1	1	0	1
7	0	1	1	1	1	1	0

	1	2	3	4	5	6	7
1	0	1	1	1	1	1	0
2	1	0	1	1	0	1	0
3	1	1	0	1	0	1	1
4	1	0	0	0	0	1	0
5	1	1	1	1	0	1	0
6	0	0	1	0	0	0	0
7	1	1	1	1	1	1	0

-y +z -z

$$
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
2 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
3 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
4 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
5 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
6 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
\qquad
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
4 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
5 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
7 & 0 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
\qquad
\begin{array}{c|ccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
3 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
4 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
5 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
6 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
7 & 0 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}
$$

The optimal assembly sequences obtained for the gear assembly using developed algorithm are tabulated in the table-1. In this two optimal assembly sequences are obtained with minimum number of directional changes, gripper changes and part movement. In the part movement, generally distance has been considered as the objective constraint. As the other two objective constraints are unit less so, the part movement distance is multiplied with a large constant (e^{10}) to convert it to unit less.

Table-1: Optimal assembly sequences for the gear assembly.

SL.No	Assembly Sequence	No.of directional changes	No.of gripper changes	Part Movement	Fitness Value
1	3 2 1 4 5 7 6	2	4	1.946	2.622
2	3 2 1 5 4 6 7	2	4	1.946	2.622

A graph shown in the figure-3 is plotted between number of iterations and fitness value. In this, the minimum fitness value is obtained after 44 iterations, which is less compared to the advanced immune strategy algorithm.

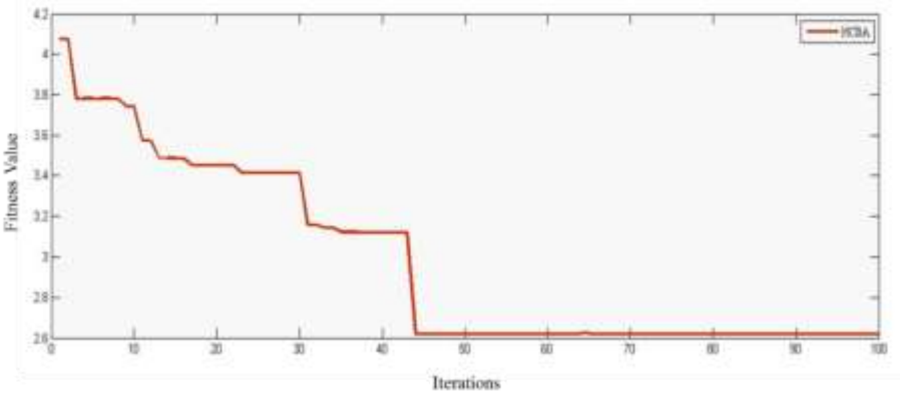

Figure-3: Graph between number of iterations and fitness value.

The results shown in the table -2 of the developed algorithm are compared with the advanced immune strategy in terms of number of optimal assembly sequences, fitness value and execution time. Out of those execution time to get the optimal assembly sequences is less compared to the advanced immune strategy.

Table-2: Comparison results

Type of Assembly	No.of sequences	Fitness value	Execution time (Sec)

Gear Assembly			
Advanced Immune Strategy [21]	2	2.622	0.64
HCBA	2	2.622	0.61

Case-2: Wall Rack Assembly

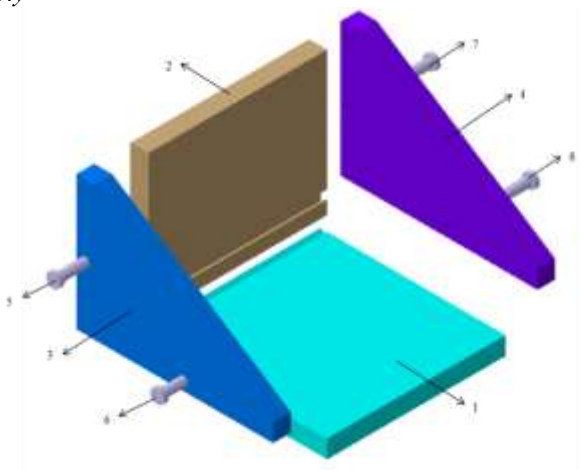

Figure -4: Wall rack assembly

In the second case, wall rack assembly shown in the figure-4 is considered to compare the results of the proposed algorithm with the algorithms like GA, ACO, GWO and HAWA. In this, two objective constraints shown in the equation (4) is considered to evaluate the quality of the sequence.

Liaison data

	1	2	3	4	5	6	7	8
1	0	1	1	1	0	1	0	1
2	1	0	1	1	1	0	1	0
3	1	1	0	0	1	1	0	0
4	1	1	0	0	0	0	1	1
5	0	1	1	0	0	0	0	0
6	1	0	1	0	0	0	0	0
7	0	1	0	1	0	0	0	0
8	1	0	0	1	0	0	0	0

Stability data

	1	2	3	4	5	6	7	8
1	0	2	1	1	0	2	0	2
2	2	0	1	1	2	0	2	0
3	1	1	0	0	2	2	0	0
4	1	1	0	0	0	0	2	2
5	0	1	1	0	0	0	0	0
6	1	0	1	0	0	0	0	0
7	0	1	0	1	0	0	0	0
8	1	0	0	1	0	0	0	0

Geometrical feasibility matrices

x+ x- y+

	1	2	3	4	5	6	7	8
1	0	0	0	1	1	0	1	1
2	1	0	0	1	0	1	1	1
3	1	1	0	1	0	0	1	1
4	0	0	1	0	1	1	1	1
5	1	1	1	1	0	1	1	1
6	1	1	1	1	1	0	1	1
7	1	0	1	0	1	1	0	1
8	0	1	1	0	1	1	1	0

	1	2	3	4	5	6	7	8
1	0	1	1	0	1	1	1	0
2	0	0	1	0	1	1	0	1
3	0	0	0	1	1	1	1	1
4	1	1	1	0	1	1	0	0
5	1	0	0	1	0	1	1	1
6	0	1	0	1	1	0	1	1
7	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	1	0

	1	2	3	4	5	6	7	8
1	0	0	1	1	1	0	1	0
2	0	0	1	1	0	0	0	1
3	1	1	0	1	0	0	1	1
4	1	1	1	0	1	1	0	0
5	1	0	0	1	0	1	1	1
6	0	1	0	1	1	0	1	1
7	1	0	1	0	1	1	0	1
8	0	1	1	0	1	1	1	0

y-

	1	2	3	4	5	6	7	8
1	0	0	1	1	1	0	1	0
2	0	0	1	1	0	1	0	1
3	1	1	0	1	0	0	1	1
4	1	1	1	0	1	1	0	0
5	1	0	0	1	0	1	1	1
6	0	0	0	1	1	0	1	1
7	1	0	1	0	1	1	0	1
8	0	1	1	0	1	1	1	0

z+

	1	2	3	4	5	6	7	8
1	0	0	1	1	0	0	0	0
2	0	0	1	1	0	1	1	1
3	1	1	0	1	0	0	1	1
4	1	1	1	0	1	1	1	1
5	1	1	1	1	0	1	1	1
6	0	1	1	1	1	0	1	1
7	1	0	1	0	1	1	0	1
8	0	1	1	0	1	1	1	0

z-

	1	2	3	4	5	6	7	8
1	0	0	1	1	1	0	1	0
2	0	0	1	1	1	1	0	1
3	1	1	0	1	1	1	1	1
4	1	1	1	0	1	1	0	0
5	0	0	0	1	0	1	1	1
6	0	1	0	1	1	0	1	1
7	0	1	1	1	1	1	0	1
8	0	1	1	1	1	1	1	0

The results of the proposed algorithm are shown in the table-3. In this, 8 optimal assembly sequences with minimum number of directional changes and gripper changes are obtained.

Table-3: Represents the assembly sequences for wall rack assembly

SI.NO	Assembly sequence								No.of directional Changes	No.of gripper Changes	Fitness value
1	1	2	3	4	7	8	5	6	2	2	2
2	1	2	3	4	7	8	6	5	2	2	2
3	1	2	3	4	8	7	5	6	2	2	2
4	1	2	3	4	8	7	6	5	2	2	2
5	2	1	4	3	5	6	7	8	2	2	2
6	2	1	4	3	5	6	8	7	2	2	2
7	2	1	4	3	6	5	7	8	2	2	2
8	2	1	4	3	6	5	8	7	2	2	2

A graph shown in the figure-5 is plotted between number of iterations and fitness values. The algorithm is run for 300 iterations; fitness value is converged after 49 iterations only.

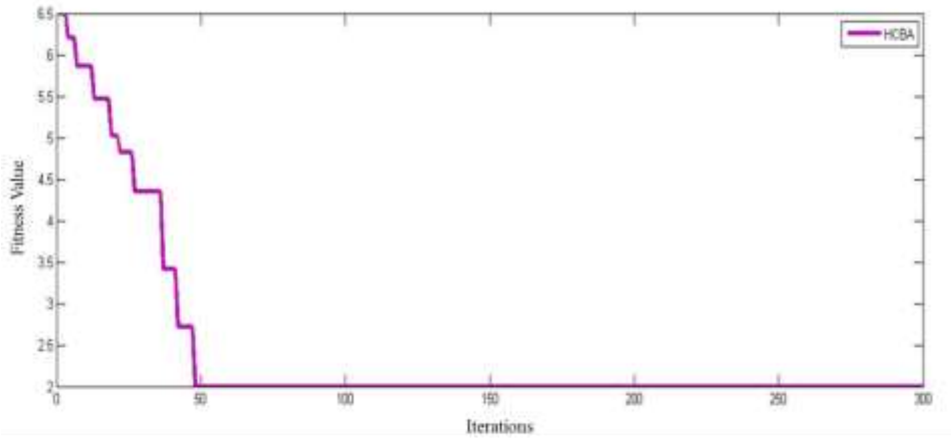

Figure-5: Graph between number of iterations and fitness value.

The results obtained from the developed algorithm (HCBA) are compared with the several well-known algorithms like GA, ACO, GWO and HAWA in terms of fitness value and CPU time, which is shown in the table-3.

Table-3: Represents the assembly sequences for wall rack assembly

Indicator	GA [8]	ACO [8]	GWO [22]	HAWA [8]	HCBA
Rack assembly					
Fitness Value	5.2727	5.4545	5.4090	5	2
Avg.CPU time(Sec)	4.7861	4.0984	5.1246	4.4258	2.9263

5. Conclusions

In this paper, a new hybrid algorithm is developed by the combination of the cuckoo search and bat algorithms respectively. Mainly in this, the searching nest in cuckoo search algorithm and updating the velocities and positions of the bats in the bat algorithm are combined to form the hybrid algorithm. The following conclusions are been observed.

1. The developed hybrid algorithm (HCBA) is able to obtain the optimal assembly sequences with less Avg. CPU time compared to the other algorithms, compared in the above section-4. Similarly it generates more optimal sequences with less fitness value compared to the other algorithms, which is shown in the section-4.
2. The algorithm is compared with two different assemblies to evaluate the quality of the solution in terms of fitness value and CPU execution time. The proposed algorithm is compared with the Advanced Immune based Strategy algorithm for gear assembly. In this, the avg. CPU time for the developed algorithm is less compared to the Advanced Immune based Strategy algorithm.

3. The developed algorithm is also compared GA, ACO, GWO and HAWA algorithms for wall rack assembly. In this, the avg. CPU time and fitness value of the proposed algorithm is less compared to the other algorithms

As a future work this algorithm can be implemented for the more number part assemblies. Moreover, the algorithm may be extended to the flexible part assemblies and the parts which are to be assemble other than principle axes.

References

1. M. V. A. R. Bahubalendruni and B. B. Biswal, (2014), "Computer aid for automatic liaisons extraction from cad based robotic assembly," IEEE 8th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, pp. 42-45.
2. Raju Bahubalendruni M.V.A., Biswal B.B. (2015) An Intelligent Method to Test Feasibility Predicate for Robotic Assembly Sequence Generation. In: Jain L., Patnaik S., Ichalkaranje N. (eds) Intelligent Computing, Communication and Devices. Advances in Intelligent Systems and Computing, vol 308. Springer, New Delhi.
3. Bala Murali, G., et al. (2017), "An Intelligent Strategy for Automated Assembly Sequence Planning While Considering DFA Concept."
4. De Fazio, T.L. and Whitney, D.E. (1987), “Simplified generation of all mechanical assembly sequences”, IEEE Journal on Robotics and Automation, Vol. 3 No. 6, pp. 640-658.
5. Homem de Mello, L.S. and Sanderson, A.C. (1990), “AND/ OR graph representation of assembly plans”, IEEE Transactions on Robotics and Automation, Vol. 6 No. 2, pp. 188-199.
6. Bala Murali G., Deepak B.B.V.L., Raju Bahubalendruni M.V.A., Biswal B.B. (2017) Optimal Assembly Sequence Planning Towards Design for Assembly Using Simulated Annealing Technique. In: Chakrabarti A., Chakrabarti D. (eds) Research into Design for Communities, Volume 1. ICoRD 2017. Smart Innovation, Systems and Technologies, vol 65. Springer, Singapore.
7. Gunji, B., et al. (2017), "Hybridized genetic-immune based strategy to obtain optimal feasible assembly sequences." International Journal of Industrial Engineering Computations Vol.8 No.3 pp.333-346.
8. Mohd Fadzil Faisae Ab Rashid, (2017) "A hybrid Ant-Wolf Algorithm to optimize assembly sequence planning problem", Assembly Automation, Vol. 37 No. 2, pp.238-248.
9. Ayoub, R.G. and Doty, K.L. (1989), “Representation for discrete assembly sequences in task planning”, Proceedings -IEEE Computer Society’s International Computer Software & Applications Conference, Orlando, pp. 746-753.
10. Chakrabarty, S. and Wolter, J. (1997), “A structure-oriented approach to assembly sequence planning”, IEEE Transactions on Robotics and Automation, Vol. 13 No. 1,pp. 14-29.
11. Xiaoming, Z. and Pingan, D. (2008), “A model-based approach to assembly sequence planning”, International Journal of Advanced Manufacturing Technology, Vol. 39 Nos 9/10, pp. 983-994.
12. Wong, H. and Leu, M.C., 1993. Adaptive genetic algorithm for optimal printed circuit board assembly planning. CIRP Annals-Manufacturing Technology, Vol. 42 No.2, pp.17-20.
13. Bonneville, F., Perrard, C. and Henrioud, J.M., (1995), October. A genetic algorithm to generate and evaluate assembly plans. In Emerging Technologies and Factory Automation, 1995. ETFA'95, Proceedings., 1995 INRIA/IEEE Symposium on Vol. 2, pp. 231-239. IEEE.
14. Dini, G., Failli, F., Lazzerini, B. and Marcelloni, F., (1999). Generation of optimized assembly sequences using genetic algorithms. CIRP Annals-Manufacturing Technology, Vol. 48 No. 1, pp.17-20.

15. Hong, D.S. and Cho, H.S., (1999). A genetic-algorithm-based approach to the generation of robotic assembly sequences. Control Engineering Practice, Vol. 7 No. 2, pp.151-159.
16. Smith, S.S.F. and Liu, Y.J., (2001). The application of multi-level genetic algorithms in assembly planning. Journal of Industrial Technology, Vol. 17 No. 4, pp.1-4.
17. Failli, F. and Dini, G., (2000), June. Ant colony systems in assembly planning: a new approach to sequence detection and optimization. In Proceedings of the 2nd CIRP international seminar on intelligent computation in manufacturing engineering pp. 227-232.
18. Wang, J.F., Liu, J.H., Li, S.Q. and Zhong, Y.F., (2003). Intelligent selective disassembly using the ant colony algorithm. AI EDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 17 No.04, pp.325-333.
19. McGovern, S.M. and Gupta, S.M., (2006). Ant colony optimization for disassembly sequencing with multiple objectives. The International Journal of Advanced Manufacturing Technology, Vol. 30 No.5-6, pp.481-496.
20. Wang, H., Rong, Y. and Xiang, D., (2014). Mechanical assembly planning using ant colony optimization. Computer-Aided Design, Vol. 47, pp.59-71.
21. M.V.A. Raju Bahubalendruni, B.B.V.L. Deepak, Bibhuti Bhusan Biswal, (2016) "An advanced immune based strategy to obtain an optimal feasible assembly sequence", Assembly Automation, Vol. 36 No. 2, pp.127-137.
22. Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014), “Grey Wolf Optimizer”, Advances in Engineering Software, Vol. 69, pp. 46-61.