Modified BAT Algorithm for Optimum Assembly Sequence Planning

G. Bala Murali *1[a], Pradip Kumar Sahu 1[b], B.B.V.L. Deepak 1[c], B.B. Biswal 1[d]
[1] Department of Industrial Design, National Institute of technology, Rourkela, Odisha, India-769008.
Email ID: bmgunji@gmail.com [a]; pradipsahu2@gmail.com [b]; bbv@nitrkl.ac.in [c]; bbbiswal@nitrkl.ac.in [d].

Abstract. Assembly Sequence Planning (ASP) is one of the important optimization problems in manufacturing. Optimum assembly sequence is difficult due to various reasons: As ASP is a NP hard combinatorial problem, achieving the optimum assembly sequence is a difficulty process. Moreover, ASP problem is multi-model optimization problem where a product can assemble in many possible ways. As the part count in the assembly increases, the time for assembly is more and obtaining the optimum assembly sequences is difficult. Many mathematical algorithms are proposed to obtain optimum solutions, which performs poorly. Meanwhile researchers are motivated towards developing the Artificial Intelligence (AI) techniques to solve ASP problems due to their less search space for implementing even the complex assemblies. The challenging task in ASP problem is automatic extraction assembly constraints to obtain the optimum assembly sequence. Keeping this thing in mind, in this paper, a Modified BAT Algorithm (MBA) has been implemented to solve ASP problem. In this paper first time BAT algorithm is applied to solve discrete optimization problem.

Keywords: Assembly sequence planning, Computer-aided Design, Metaheuristic optimization algorithms, Modified Bat Algorithm (MBA)

1. Introduction

One of the important decision making tasks in manufacturing is assembly sequence planning. Determination of assembly sequence is that the order of parts that are to be joined one after other to make assembly. The assembly not only affects the time of the manufacturing but also increase the cost of manufacturing. Assembly planning directly or indirectly involves many operations like fixture arrangement, tool changes between two consecutive operations etc. Different sequences are possible for a parts during assembly. However, making choice is a difficulty process for getting optimum assembly sequence because assembly sequence planning problem mainly three reasons: Initially random generated sequence has to test the feasibility, extracting the absolute constraints is difficulty process. Secondly, with increase in the part number the number of possible may increases drastically, getting an optimal sequence out of those possible sequence is difficulty process and consumes lot of time. Thirdly, forming an objective function for generating the quality assembly sequence involves lot of complexity especially for the complex geometry part assembly. At the initial stages mostly they used conventional mathematical models to generate the optimal solution, but they perform poorly when comes to large part count assemblies. Later soft computing techniques are developed to generate optimal assembly sequence, which suits best for multi-objective optimization problems [1-3]. In this paper modified BAT algorithm is proposed along with the automatic extraction of absolute constraints to test the feasibility of the sequence. The algorithm is compared with different well known algorithms like Genetic Algorithm (GA), Ant Colony Optimisation (ACO) algorithm, Improved Harmonic Search (IHS) algorithm and Flower Pollination Algorithm (FPA). The proposed algorithm shows the better results than the compared algorithms.

2. Literature review

Assembly is the process of joining the parts one after the other in an order. To make assembly give the order of components (called it as sequence) has to be generated, which is known as assembly sequence planning. For example, if a product is having 'n' number of parts/ components are to be join in an sequential order say component 1, followed by 2 and so on…, the assembly sequence is represented by [1,2,3….]. To generate this type of sequence, first we have to check the feasibility of the sequence, later on develop the fitness equation using objective constraints to evaluate quality of the generated sequence. To generate the optimal assembly sequence AI techniques are being used, in this Hybrid Algorithms (HA) [4] plays prominent role to solve complex assemblies, where individual algorithms fails. In the rapid growth world to reduce the manufacturing cost Design for Assembly (DFA) concept has been introduced to get the modified topology of the products [5]. Even though the product obtained from DFA concept had involve complexity in solving the ASP problems.

2.1. Assembly sequence planning and optimization problem
Assembly sequence planning is multi objective optimization problem. For getting optimal assembly sequence at a time more than two assembly constraints are to used. Generally, assembly constraints are two type: one is absolute constraint, which is useful to check the feasibility of the sequence. The other objective constraint, used to get the quality solution.

3. Proposed methodology for assembly sequence optimization using modified BAT algorithm
In this research work a modified BAT algorithm is proposed to generate quality assembly sequence. In this algorithm, loudness and pulse emission rate of the bats has been updated for every iteration so that the assembly sequence generated will have more quality compared to other algorithms. More over for this algorithm results will converge in less number of iterations compared to other algorithms.

3.1. Assembly Constraints to Check the Feasibility of the Solution
Assembly constraints are generally two types: one is absolute constraints, used for checking the feasibility of the assembly sequence. Second one is optimization constraints, used for evaluating the fitness of the sequence for generating the quality solution.

The absolute constraints are generally liaison data, geometrical feasibility data, stability data and mechanical feasibility data. The required matrices of the absolute constraints are extracted from CATIA V5 R17 using macros. The absolute constraints data of motor drive assembly shown in the Fig. 1. is as follows:

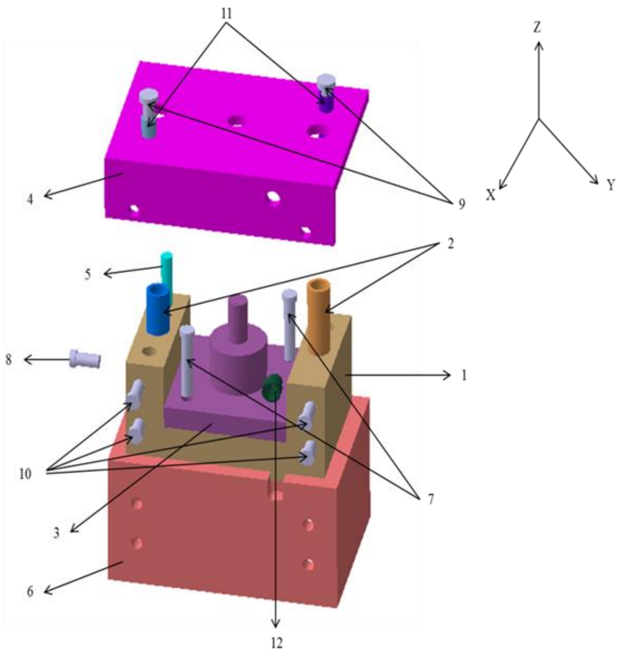

Fig. 1. DeFazio Motor drive assembly [6].

Table 1. Tools and part names of the motor drive assembly

Part no.	Part name	Tool/gripper name
1	Motor base	Parallel gripper
2	Bushing	Parallel gripper
3	Motor	Parallel gripper
4	End plate	Adaptive gripper
5	Sensor	Adaptive gripper
6	Cover	Parallel gripper
7	Motor screw	Screw driver
8	Set screw	Allen key
9	End plate screw	Screw driver
10	Cover screw	Screw driver
11	Stand off	wrench
12	Grommet	Hammer

Liaison data matrix:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	0	0	0	0	1	1	1	1	0
2	1	0	1	1	1	1	1	1	1	0	0	0
3	1	1	0	0	0	0	1	0	0	1	0	0
4	0	1	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	1	0	0
6	0	1	0	0	0	0	0	0	0	1	0	1
7	0	1	1	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0	0	0
10	1	0	1	0	1	1	0	0	0	0	1	1
11	1	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	1	0	0	0	1	0	0

Stability data matrix:

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	2	2	0	0	0	0	2	2	2	2	0
2	1	0	2	2	2	2	2	2	2	0	0	0
3	1	1	0	0	0	0	2	0	0	2	0	0
4	0	1	0	0	0	0	0	0	0	0	0	0
5	0	1	0	0	0	0	0	0	0	1	0	0
6	0	1	0	0	0	0	0	0	0	1	0	2
7	0	1	1	0	0	0	0	0	0	0	0	0
8	1	1	0	0	0	0	0	0	0	0	0	0
9	1	1	0	0	0	0	0	0	0	0	0	0
10	2	0	2	0	2	2	0	0	0	0	2	2
11	1	0	0	0	0	0	0	0	0	1	0	0
12	0	0	0	0	0	1	0	0	0	1	0	0

In liaison data matrix '1' represents contact between the parts and '0' represents no contact between the parts. In stability data matrix '0' represents no stability, '1' represents partial stability and '2' represents the permanent stability due to self-alignment.

Geometrical feasibility data:

X+

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	1	1	1	0
2	0	0	0	0	0	1	0	0	1	1	1	1
3	0	0	0	1	0	1	0	1	1	0	1	1
4	0	0	1	0	1	1	1	1	1	1	1	1
5	0	0	1	1	0	1	1	1	1	0	1	1
6	0	1	1	1	1	0	1	1	1	0	1	0
7	0	0	0	1	1	1	0	1	1	1	1	1
8	0	0	0	1	1	1	1	0	1	1	1	1
9	0	0	1	1	1	1	1	1	0	1	1	1
10	1	1	0	1	0	0	1	1	1	0	0	0
11	1	1	1	1	1	1	1	1	1	0	0	1
12	0	1	1	1	1	0	1	1	1	0	1	0

X-

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	1	1	0
2	0	0	0	0	0	1	0	0	0	1	1	1
3	0	0	0	1	1	1	0	0	1	0	1	1
4	0	0	1	0	1	1	1	1	1	1	1	1
5	0	0	0	1	0	1	1	1	1	0	1	1
6	0	1	1	1	1	0	1	1	1	0	1	0
7	0	0	0	1	1	1	0	1	1	1	1	1
8	0	0	1	1	1	1	1	0	1	1	1	1
9	1	1	1	1	1	1	1	1	0	1	1	1
10	1	1	0	1	0	0	1	1	1	0	0	0
11	1	1	1	1	1	1	1	1	1	0	0	1
12	0	1	1	1	1	0	1	1	1	0	1	0

Y+

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	1	0	1	1	0
2	0	0	0	0	0	1	0	1	0	1	1	1
3	0	0	0	1	1	1	0	1	1	0	1	1
4	0	0	1	0	1	1	1	1	1	1	1	1
5	0	0	1	1	0	1	1	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	0	1	0
7	0	0	0	1	1	1	0	1	1	1	1	1
8	0	0	1	1	1	1	1	0	1	1	1	1
9	0	0	1	1	1	1	1	1	0	1	1	1
10	0	0	0	0	0	0	1	1	1	0	1	0
11	0	1	1	1	1	1	1	1	1	0	0	1
12	0	1	1	1	1	0	1	1	1	0	1	0

Y-

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	0	0	0	0	1	1
3	0	0	0	1	1	1	0	1	1	0	1	1
4	0	0	1	0	1	0	1	1	1	0	1	1
5	0	0	1	1	0	1	1	1	1	0	1	1
6	0	1	1	1	1	0	1	1	1	0	1	0
7	0	0	0	1	1	1	0	1	1	1	1	1
8	1	1	1	1	1	1	1	0	1	1	1	1
9	0	0	1	1	1	1	1	1	0	1	1	1
10	1	1	0	1	0	0	1	1	1	0	0	0
11	1	1	1	1	1	1	1	1	1	1	0	1
12	0	1	1	1	1	0	1	1	1	0	1	0

Z+

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	1	1	1	1	0	0	0	1	1
2	1	0	0	0	0	0	0	0	0	0	1	0
3	1	1	0	1	1	1	0	1	1	0	1	1
4	1	1	1	0	1	1	1	1	1	0	1	1
5	1	1	1	1	0	1	1	1	1	1	1	1
6	1	1	1	1	1	0	1	1	1	1	1	0
7	1	1	1	1	1	1	0	1	1	0	1	1
8	0	0	1	1	1	1	1	0	1	0	1	1
9	0	0	1	1	1	1	1	1	0	0	1	1
10	1	1	1	1	1	1	1	1	1	0	0	0
11	1	1	1	1	1	1	1	1	1	0	0	1
12	1	1	1	1	1	1	1	1	1	1	1	0

Z-

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	1	1	1	1	1	0	0	1	1	1
2	1	0	1	1	1	1	1	0	0	1	1	1
3	1	0	0	1	1	1	1	1	1	1	1	1
4	1	0	1	0	1	1	1	1	1	1	1	1
5	1	0	1	1	0	1	1	1	1	1	1	1
6	1	0	1	1	1	0	1	1	1	1	1	1
7	1	0	0	1	1	1	0	1	1	1	1	1
8	0	0	1	1	1	1	1	0	1	1	1	1
9	0	0	1	1	1	1	1	1	0	1	1	1
10	0	0	0	0	1	1	0	0	0	0	0	1
11	1	1	1	1	1	1	1	1	1	0	0	1
12	1	0	1	1	1	0	1	1	1	0	1	0

Geometrical fesibility matrix gives the information about the fesibility of part to assembly the in a particluar direction. Total six matrices will be generated for six directions to test the fesibiltiy of the part during assembly. In the matrix ‘0’ represents not feasible to join the part in that direction and ‘1’ represenst feasible to join the part in that direction.

3.2. Modified BAT Algorithm

BAT algorithm was initially proposed by Yang [7], inspired by echolocation behaviour of micro bats. This algorithm is used for achieving the global optimum solution. Even though it achieves global solution, but for the discrete optimization problems like ASP problems, the original form of the algorithm is not suitable. So, in this paper a Modified BAT Algorithm (MBA) has been proposed to achieve global optimal solution. The detailed flow chart of the algorithm is shown in the Fig. 2.

In MBA for every iteration pulse emission rate (r) and loudness (A) are updated so that the solution accuracy will increase. The proposed algorithm is applied on 12 part motor drive assembly shown in the Fig. 1.

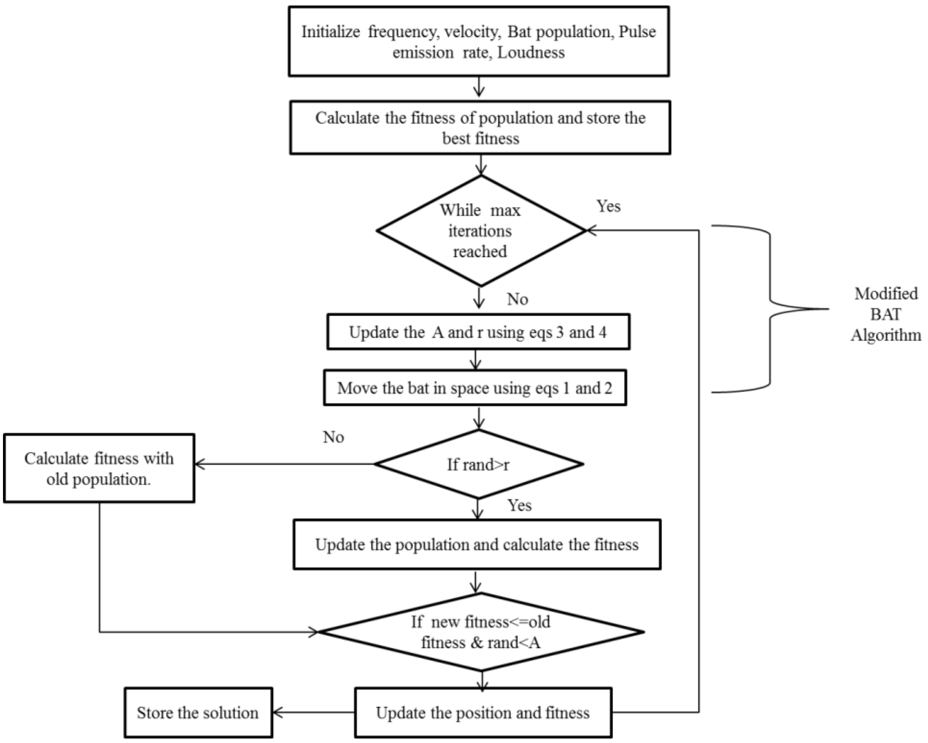

Fig. 2. Modified BAT algorithm

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta \tag{1}$$

$$v_i^{t+1} = v_i^t + \left(x_i^{t-1} - x*\right)f_i \tag{2}$$

Where β ranges from [0-1], f_{min}=0 &f_{max} =100, $x*$ is the current global best solution , t= number of iterations.

$$A_i^{t+1} = a * A_i^t \tag{3}$$

$$r_i^t = r_i\left(1 - e^{(-\gamma t)}\right) \tag{4}$$

Where A and r are loudness and pulse emission rate, a& γ are constants.

To evaluate the quality of the generated sequence, a fitness equation is developed by considering Gripper Changes (G.C) and Directional Changes (D.C) as optimization constraints. The fitness equation shown (6) has developed by giving equal priority to the both constraints.

$$f = w_1 * D.C + w_2 * G.C \tag{5}$$

where, $w_1 = w_2 = 0.5$

4. Results and Discussion

Modified BAT algorithm is implemented in this paper to obtain the optimum assembly sequence for the motor drive assembly. In this a population size of 50 is considered and the algorithm is run for 500 iterations. The convergence graph with a fitness value of 0.2 is shown in the Fig. 3 obtained after 300 iterations, which is less compared to Flower pollination algorithm (FPA) [8].

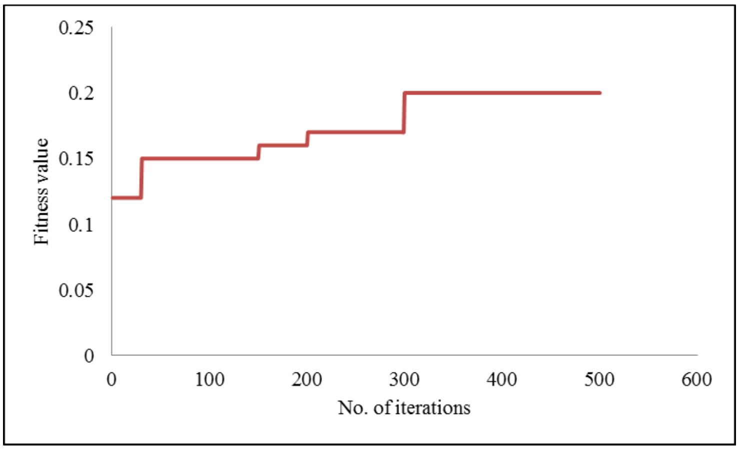

Fig. 3. Convergence graph of MBA for motor drive assembly for 500 iterations.

Table 2: List of optimum assembly sequences for motor drive assembly, which is run for 500 iterations and a population size of 20.

Sl.No	Optimal assembly sequence	No. of directional changes	No. of tool changes	Fitness value
1	[1 2 3 5 8 11 7 4 9 12 6 10]	5	9	0.200
2	[1 2 3 7 5 11 8 4 9 12 6 10]	5	9	0.200
3	[1 2 3 11 5 7 8 4 9 12 6 10]	5	9	0.200
4	[1 3 2 5 7 8 11 4 9 12 6 10]	5	9	0.200
5	[1 3 2 5 7 11 8 4 9 12 6 10]	5	9	0.200
6	[1 3 2 5 8 7 11 4 9 12 6 10]	5	9	0.200
7	[1 3 2 5 11 8 7 4 9 12 6 10]	5	9	0.200
8	[1 3 2 11 5 8 7 4 9 12 6 10]	5	9	0.200
9	[1 3 2 11 7 5 8 4 9 12 6 10]	5	9	0.200
10	[1 3 2 5 8 11 7 4 9 12 6 10]	5	9	0.200

The unique optimum assembly sequences obtained for population size of 20 are 10. The algorithm is run for 500 iterations and results of the assembly sequences along with number of gripper/tool changes and directional changes of the assembly sequence is tabulated in Table 2.

4.1. Comparison of results

The proposed algorithm results are tabulated in Table 3 and compared with the different algorithms like GA [1], ACO [10], IHS [9] and FPA [8]. By the results comparison the proposed algorithm performs better than the remaining algorithms.

Table 3: Comparison of results of different algorithms for motor drive assembly for 500 iterations.

	GA[1]	ACO[10]	IHS[9]	FPA[8]	MBA
Optimum fitness value	0.2	0.2	0.2	0.2	0.2
Optimum tool changes	9	9	9	9	9
Optimum directional changes	5	5	5	5	5
Number of optimum assembly sequences	4	9	8	9	10

5. Conclusion

In this paper, modified BAT algorithm is proposed to obtain the optimum assembly sequence. The algorithm is implemented on 12 part motor drive assembly and the results are compared with the various well known algorithms like GA, ACO, IHS and FPA. The results obtained from the modified BAT algorithm are better than the compared algorithms. Moreover, the optimum unique solutions obtained are more compared to other algorithms with in less number iterations.

References

[1] Chen, Shiang-Fong, and Yong-Jin Liu. "An adaptive genetic assembly-sequence planner." International Journal of Computer Integrated Manufacturing 14.5 (2001): 489-500.
[2] Choi, Young-Keun, Dong Myung Lee, and Yeong Bin Cho. "An approach to multi-criteria assembly sequence planning using genetic algorithms." The International Journal of Advanced Manufacturing Technology 42.1 (2009): 180-188.
[3] Wang, Hui, Yiming Rong, and Dong Xiang. "Mechanical assembly planning using ant colony optimization." Computer-Aided Design 47 (2014): 59-71.
[4] Gunji, B., Deepak, B., Bahubalendruni, M & Biswal, B. (2017). Hybridized genetic-immune based strategy to obtain optimal feasible assembly sequences.International Journal of Industrial Engineering Computations, 8(3), 333-346.
[5] 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.
[6] Boothroyd, Geoffrey. Assembly automation and product design. CRC Press, 2005.
[7] Gandomi, Amir H., and Xin-She Yang. "Chaotic bat algorithm." Journal of Computational Science 5.2 (2014): 224-232.
[8] Mishra, Atul, and Sankha Deb. "Assembly sequence optimization using a flower pollination algorithm-based approach." Journal of Intelligent Manufacturing (2016): 1-22.
[9] Li, Xinyu, et al. "Assembly sequence planning based on an improved harmony search algorithm." The International Journal of Advanced Manufacturing Technology 84.9-12 (2016): 2367-2380.
[10] Wang, J. F., J. H. Liu, and Y. F. Zhong. "A novel ant colony algorithm for assembly sequence planning." The international journal of advanced manufacturing technology 25.11 (2005): 1137-1143.