

PAPER • OPEN ACCESS

A case study of energy efficient assembly sequence planning problem

To cite this article: M A Abdullah *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **469** 012013

View the [article online](#) for updates and enhancements.

Recent citations

- [Improved moth flame optimization algorithm to optimize cost-oriented two-sided assembly line balancing](#)
Mohd Fadzil Faisae Ab. Rashid *et al*



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the **collection** - **download the first chapter of every title for free.**

A case study of energy efficient assembly sequence planning problem

M A Abdullah¹, M F F Ab Rashid¹, Z Ghazali¹ and A N M Rose¹

¹Faculty of Mechanical Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

*Corresponding author's e-mail: arifabdullah22@gmail.com

Abstract. Energy efficiency has become an important issue in manufacturing industry, since it is one of the biggest energy consumers in the world. Despite the importance of energy efficiency, it is much obvious that the research in assembly sequence that focus on environmental aspect is still lacking. In Assembly Sequence Planning (ASP), the research on problem optimization is mainly demanded for the effective computational approach to determine the best assembly sequence. This paper presents a case study from an electronic product assembly that considers the energy utilization during assembly process. In particular, the case study focuses to reduce the idle energy utilization in assembly process. The case study was optimized using newly proposed Moth-Flame Optimization (MFO) and then being compared with well-frequent used algorithms including Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA). The result of the computational experiment test was divided into comparison of assembly layout between MFO proposed layout and existing layout. Besides, the statistical test involving Analysis of Variance (ANOVA) and post-hoc test of Fisher's Least Significant Difference (LSD) were then conducted. The proposed MFO performed better in terms of the best minimum fitness (0.401681), average fitness (0.415308), standard deviation of fitness (0.022601), with appropriate computational time and power consumed. In meantime, the results also indicated that the case study was suitable in the development of energy efficient model for ASP.

1. Introduction

Assembly sequence planning (ASP) become one of the significant complex NP-hard combinatorial optimization problems, which brings attention in industry to reduce the total manufacturing cost by minimize the assembly time and energy [1]. One of the main challenges in product design and manufacturing was ASP. With involvement of many product components or parts, ASP problem was highly difficult to solve [2]. Since different sequences affects in different tooling arrangements, assembly sequence planning (ASP) is notified as the indivisible analysis part performed during initial phases of product and process design. Hence, the assembly cost, productivity and quality are directly affected by the different sequences [3]. From the possible different assembly sequences, ASP is needed to obtain the optimum assembly sequence [1]. To identify the best optimum assembly sequence, ASP optimization is demanded for effective computational approach due to the complexity of problem. In ASP optimization, the initial prominent step is to construct a representation method scheme to model the assembly problem. The modeling of assembly problem will directly impact the effectiveness of an assembly plan.

The ASP representation schemes are developed to fit with their optimization problem characteristics and attributes. These modeling approaches are specified into three major categories; part-based, task-



based and connector-based representation. According to the established research, the best accurate approach for ASP representation is by modelling base on the part of product [4]. This approach contributes to greater number of assembly sequence possibility which leads to harder optimization process. A simpler version model is then proposed, whereby the base part of product is described prior to development of the precedence constraint. The second modeling approach is notified as task-based which represents assembly task in the assembly process, instead of assembly part. During the task-based modeling process, the assembly direction needs for the definition of moving and fixed part. To conduct this type of modeling process, it must be minimum of two components for an assembly task. The least frequent modelling approach implemented in ASP optimization is referred as connector-based modeling. The appropriate optimization data and precedence relations are specified as to establish assembly model from connector structure before ASP. A connector probably be utilized to assemble more than one component. Since connector is the only modeled element, this approach stands high chance to minimize the assembly problem complexity.

The meta-heuristics soft computing approaches consisting of optimization algorithms show their predominance in solving ASP problems, which contradicted with the conventional techniques. Various prominent meta-heuristics optimization algorithms involving ant colony optimization algorithm (ACO), genetic algorithm (GA), and particle swarm optimization algorithm (PSO) have been applied through existing studies on ASP [5]. With the major attributes of ACO algorithms that are distributed computation, positive feedback and application of constructive greedy heuristic search, this type of algorithm is utilized to develop and optimize assembly sequences of mechanical products [6]. Due to its ability which easily operate large search spaces, flexibility in determining the constraints and derive in a fitness function [7]. By using directed stochastic search of product's solution space of potential assembly plan, GA attempt to find optimal assembly plan [8]. As the platforms to find the best position in the search space, the social nature of interaction and communication of fish schooling, bird flocking and swarming theory are stimulated by the PSO algorithm [9]. The principle and parameters of PSO algorithm greatly affects the premature convergence and stochastic stagnation performance, despite having positive attributes to solve ASP problems [5]. The consideration of sustainable manufacturing and ergonomic factors will the new directions, which should be emphasized in ASP research. Hence, due to energy utilization aspect is not considered in the original optimization problem, then the assembly tool, usage time and power rating consumption for the tool are randomly developed.

Apart from problem representation, the different optimization objectives are also become the focus aspect in solving ASP problem. Based on a literature survey, minimization of direction changes and tool are the highly frequent applied optimization objectives. In ASP, the assembly direction generally is considered in six significant directions $\pm x$, $\pm y$ and $\pm z$ through the work. The assembly direction in workstation k , $k = 1, 2, \dots, m$, are divided into two mentioned categories below: (1) Each time when the assembly changes in 180° , the assembly direction changes in 180° is documented as D_{1k} , then $D_{1k} = D_{1k} + 1$. (2) The assembly direction changes in 90° is reported as D_{2k} , as whenever the assembly direction changes in 90° , thus $D_{2k} = D_{2k} + 1$ [10]. Through the case for minimization of tool change in ASP optimization, some of the research works that applied this objective are [11], [12]. Besides these two frequent used optimization objectives, another ASP objective implemented are to minimize assembly operation type [13], [14] and assembly complexity [15]. A new objective has been proposed in optimize ASP problem which is to minimize assembly power consumption. The power rating (W) input will then be utilized to calculate energy utilization (EI) in the assembly process. Industrial case study is used as the scope for energy efficient in ASP.

2. Case Study Problem

In this research, the purpose of case study is to demonstrate the actual assembly problem in industry field using the same computational experiment procedure that have been applied previously into assembly test problems. The data of case study is taken from an electronics manufacturing factory. The data collection was made during regular working hours. During the data collection process, the raw data of case study assembly was taken from an assembly line. There were three repetitions of data performed towards the product assembly. For the optimization purpose, the size of population for each algorithm was set to 20. In each run, the maximum iteration was specified as 300.

The case study problem consists of 23 assembly parts, which are marked in numbering sequence as shown in Figure 1. The assembly directions are along the X and Y axes, for every part involved. Suppose that the assembly tool and time to assemble the base part P1 are tool (T₁, T₃, T₂, and T₆), while time (39s, 73s, 64s, and 58s) respectively. Engineering information involving the assembly parts, assembly directions, assembly time usage, and assembly power are documented in Table 1. For assembly time, the running time during assembly process is recorded as 490 seconds.

The suitable tools and connections for assembling each part in this case study are also shown in Table 1 and other assembly process constraints being excluded. Besides, the ASP information including design variables, constraint and optimization parameters can be visualized through precedence graph and data matrix.

Table 1. Assembly data for case study assembly problem.

Part	Direction	Tool	Time (s)	Power (W)
P1	-z	T ₁	39	120
P2	-z	T ₁	31	120
P3	-z	T ₂	48	380
P4	+y	T ₂	89	380
P5	+y	T ₃	27	420
P6	-z	T ₃	48	420
P7	-z	T ₃	73	420
P8	-x	T ₂	21	380
P9	+y	T ₂	64	380
P10	-y	T ₄	12	100
P11	-z	T ₄	15	100
P12	-y	T ₅	65	350
P13	-y	T ₆	58	240
P14	-y	T ₇ , T ₈	60	1570
P15	+x	T ₇ , T ₈	45	1570
P16	+x	T ₉	32	0
P17	+y	T ₁₀	27	0
P18	-z	T ₁₁	70	2800
P19	-z	T ₁₂	22	70
P20	-x	T ₁ , T ₁₃	44	120
P21	-x	T ₁₄	19	750
P22	-z	T ₁₄	20	750
P23	-z	-	34	-

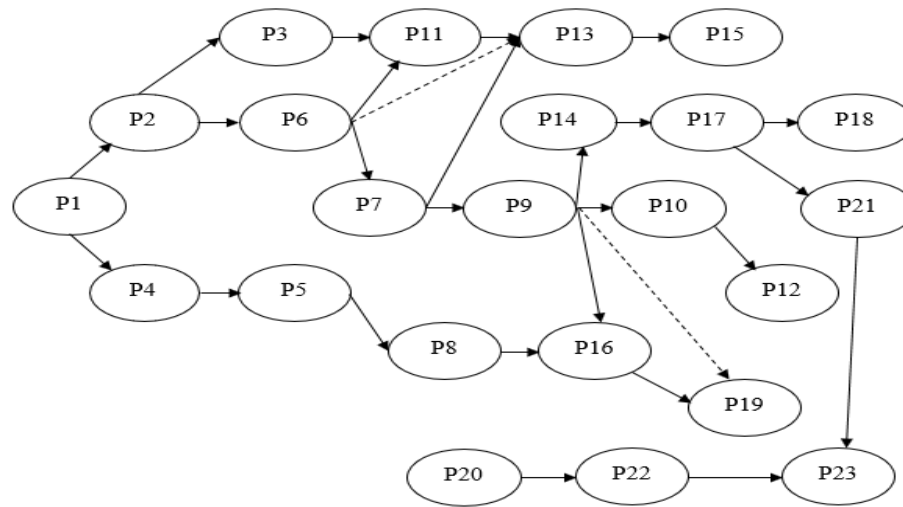


Figure 1. Precedence diagram for case study problem.

3. Moth-Flame Optimization (MFO) Algorithm

Moth-flame optimization (MFO) algorithm is one of the recent generated population-based developed algorithms firstly introduced by Mirjalili in 2015 [16]. MFO is referred as a novel nature-inspired method that based on transverse orientation of moths in space from navigating mechanism of moths [17]. With respect to the Moon, transverse orientation acquires efficient mechanism of travelling by utilizes constant angle which moves in straight line and fly spirally around the lights which converges, despite the availability of artificial light [18]. Moth is regarded as the candidate solution and variables are the position of moth in space [19]. Both of the solution differs in their treatment method and updating in every iteration [20].

During searching the search space, flames had been considered as flags that dropped by moths. Therefore, flames are observed as the best position gained so far. Using this method, a moth tends to not lose its best solution. To prevent local optimum stagnation, a designated flame is provided to every moth. The establishment of MFO flowchart brings more understanding towards the detailed computational process and mechanism of the algorithm [21]. It has been concerned that MFO algorithm is also part of the optimization, which involve searching the optimal values of control variables in minimizing the objective function. With converting their position vectors, the moths able to fly hyper dimensional space [17].

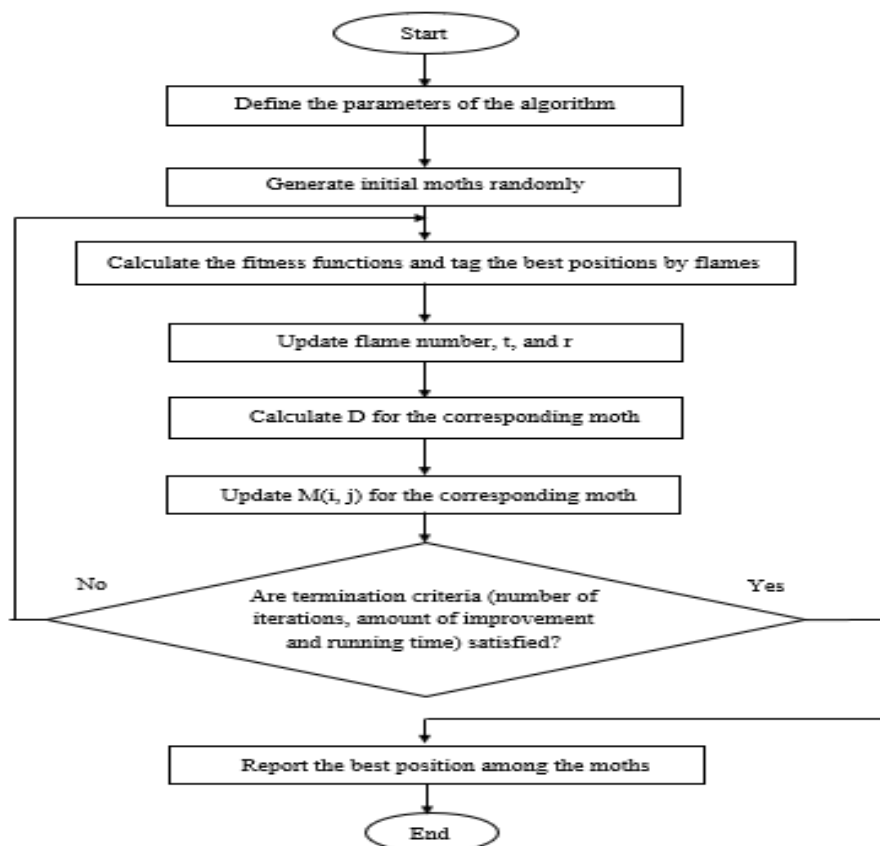


Figure 2. Flowchart of the MFO algorithm

4. Result and Discussion

In the case study, this section is categorized into two prominent main parts that including comparison of assembly layout on MFO proposed layout with the existing layout and statistical test of case study problem. The purpose of comparison is to specify the improvement or modification made between these layouts. Meanwhile, the statistical test is performed to analyze the optimization results obtained using specific analysis testing and method. For computational experiment, the optimization is repeated 20 time to reduce the pseudo-random effect. The population size is 20, while the maximum iteration is set to 300.

Table 2 shows the optimization results for the case study on energy efficient of ASP problem. The result is demonstrated in significant categories consisting of minimum fitness, maximum fitness, average fitness, standard deviation of fitness, and average CPU time. The best result for each problem is in bold italics. In this case study, all algorithms able to achieve minimum fitness. MFO algorithm obtained the best minimum fitness, maximum fitness and average fitness for all runs, which every moth search around the flame and update it to find a better solution. Hence, a moth keeps the best solution using this mechanism [19]. Besides, MFO algorithms also acquires the good standard deviation of fitness as compared to other algorithms. With respect to different locations in the search space, the update position of moths could potentially reduce the exploitation of best significant solutions [22]. Since the running time of optimization for each algorithm are different, the best average CPU time was belonged to MFO algorithm, followed by GA, PSO and ACO algorithms. Given the high convergence accuracy and global optimization ability, these positive attributes did contribute for MFO to achieve the best average CPU time. It was probably impossible to avoid the interruption during the optimization process as to get the accurate CPU time.

Table 2. Computational Experiment of Case Study.

Indicator		ACO	GA	PSO	MFO
Case Study Problem (23 parts)	Minimum fitness	0.404693	0.492716	0.491929	0.401681
	Maximum fitness	0.476879	0.591107	0.624055	0.453277
	Average fitness	0.43472	0.542075	0.562882	0.415308
	Standard deviation of fitness	0.024009	0.024597	0.032096	0.022601
	Average CPU time (s)	174.5356	62.25228	85.92661	58.87223

4.1. Comparison of Assembly Layout on MFO Proposed Layout with Existing Layout

Table 3 presents the best solution identified for case study problem on each algorithm involved. The best solution of MFO algorithm is shown in bold italic. For better explanation, the performance of MFO proposed layout was compared to existing layout of assembly objective functions as shown in Table 4. All algorithms came out with the same objective values, although there was a difference in the assembly layout. The layout of ACO was chosen since this algorithm been specified as the second best optimization algorithm after MFO. The assembly direction change is counted when the next part is assembled in different direction with the current direction. In addition, the assembly tool direction is also counted in the similar way. For the part that needs more than one tools in each algorithm, the tool changes is counted for every tool involved. There are two tools changes can be identified, for examples from P13 to P14 in the proposed MFO algorithm. For MFO layout, the number of direction change is 11, while the tool change is 19.

Table 3. Best Solution for Case Study Problem.

Algor ithm	Direction Change, <i>DC</i>	Tool Change, <i>TC</i>	Idle Energy Utilization, <i>EI</i> (W.h)	Assembly Sequence
ACO	13	22	106.5381	20 1 2 3 4 5 6 7 11 8 9 10 12 13 14 15 16 19 17 21 22 23 18
GA	17	22	147.8816	20 1 2 6 11 3 4 5 7 9 10 8 12 13 15 14 17 16 19 21 22 23 18
PSO	19	23	120.6245	20 1 4 5 8 2 3 6 7 11 9 10 12 13 14 15 16 19 17 21 22 23
MFO	11	19	111.6245	20 1 2 3 4 5 6 7 9 8 10 11 12 13 14 15 16 19 17 18 21 22 23

Table 4. Comparison of Assembly Layout.

Objective Functions	Assembly Layout	
	Existing Layout	MFO Proposed Layout
Number of Direction Change	13	11
Number of Tool Change	22	19
Idle Energy Utilization	106.5381 W.h	111.6245 W.h

4.2. Comparison of Assembly Layout on MFO Proposed Layout with Existing Layout

The overall changes for both assembly indicators are summed up as 30. On the other hand, the number of direction change is 13 and the tool change is 22 in existing layout, leads to total changes of 35. Meanwhile, the existing layout obtain lower idle energy utilization that is better than the proposed layout. Comparison of the layout between the algorithms, the proposed layout had better assembly layout than ACO algorithm which represents the existing layout. For this case study problem, the optimization result indicates that MFO managed to acquire the best minimum fitness, maximum fitness, average fitness, standard deviation of fitness, and average CPU time. Since there were positive responses occurred, the assembly usage time and power consumption able to be minimized. Therefore, this minimization helps to increase the production output yet brings more profit. Instead of secured best fitness values, the optimization objectives did not necessarily guarantee to have the best value in all indicators. Instead, MFO had the minimum direction and tool changes, however not the best on energy utilization. Possibly, this is due to the normalizing effect and power usage consumed. A small change in direction and tool changes do brings larger consequences on the fitness, since the range of energy utilization was greater than these two objective functions.

4.3. Statistical Test of Case Study

From the optimization results obtained, the statistical test involving analysis of variance (ANOVA) and post-hoc testing must be performed for further validation between each algorithm in the case study. Table 5 presents the one-way ANOVA of single factor result for case study. This result is specified into two sources of variation consisting of between groups and within groups, which also include the sum of degrees (SS), degrees of freedom (df), mean square (MS), F (distribution), P-value, and F critical. Among the two sources of variation in one-way ANOVA, the variance within groups obtained the best sum of squares and mean squares through the analysis process. When the sum of squares is lower, the mean square subsequently became lower.

Table 5. One-Way ANOVA (Single Factor) for Case Study.

Case Study Problem	Source of Variation	Sum of Squares (SS)	Degrees of Freedom (df)	Mean Square (MS)	F	P-value	F critical
	Between groups	0.22998	3	0.0767	85.028	3.2E-24	2.7249
	Within groups	0.06852	76	0.0009			

However, the degrees of freedom for within groups variance in case study was exceptionally higher, due to total number of observations contained in all data cells. Despite minus the degrees of freedom lost during the analysis because the cell means are set, it is remaining higher. The F ratio value computed in this case study was higher than F critical value, at significant level (0.05). Since the F ratio is much larger than critical value, the null hypothesis of equal population means can be rejected. Therefore, it can be concluded that there is statistically significant difference among the population means.

Table 6 indicates the post-hoc fisher's least significant difference (LSD) analysis result for case study. Larger F ratio brings smaller P-value. Given that the P-value was less than significant level α ($P \leq 0.05$), the null hypothesis H_1 needs to be rejected. Hence, the statistically significant difference does exist. Since there are differences in means, the post-hoc analysis testing must be performed. The smallest significant value between the means calculated using LSD method was 0.018911, which enable direct comparisons between the means been made.

On the other hand, Table 7 presents the mean and absolute mean difference values for case study problem which derived from analysis of LSD. As the mean represent average value of each algorithm, the difference between absolute mean value were compared to LSD value. The absolute means are considered significantly different, if the difference is larger than LSD by using proposed MFO algorithm as the benchmark algorithm. However, the differences in each algorithm was found to be lower than the LSD value. Therefore, the absolute means were considered not significantly different at $P \leq 0.05$. This

condition proves that LSD makes no adjustment or modification, for the fact that multiple comparisons are being conducted.

Table 6. Post-Hoc Fisher's Least Significant Difference (LSD) Analysis Result for Case Study.

Case Study Problem	Significant level, α	Degrees of Freedom, df	Mean Square (within), MSW	t-critical	1/n1	LSD
	0.05	76	0.000902	1.99167	0.1	0.018911

Table 7. Mean and Absolute Mean Difference of LSD Analysis Result for Case Study.

Case Study Problem	Mean				Absolute Mean Difference		
	MFO	ACO	GA	PSO	ACO	GA	PSO
	1.03386	1.03703	1.02469	1.01553	0.00317	0.00917	0.01833

5. Conclusion

This paper modeled the case study of energy efficiency for assembly sequence planning (ASP) problem, along with number of direction changes and tool changes. The comparison of assembly layout on MFO proposed layout with existing layout was made, as to show the differences occurred in terms of objective functions used. The calculation of energy utilization using MFO algorithm was performed in this case study. In this case study, the energy utilization (EI) was considered during idle mode for assembly process. A newly proposed Moth-Flame Optimization (MFO) algorithm been used, which then compared with frequently applied algorithms in the optimization for ASP. An industrial case study was implemented for computational experiment. According to the optimization results, MFO algorithm managed to achieve the best minimum fitness, maximum fitness and average fitness. Besides, the computational time for MFO was also acceptable. Furthermore, MFO obtained the good standard deviation of fitness compared to other algorithms. From this paper, the finding also summarized that energy efficient can be implemented for optimization of ASP since it is still lack in the research. The proposed MFO had huge potential to be further discovered in optimized different ASP problem.

Acknowledgement

The authors would like to acknowledge the Universiti Malaysia Pahang for supporting this research under the RDU180331 grant.

References

- [1] B. Deepak, G. Bala Murali, M. R. Bahubalendruni, and B. Biswal, *Assembly sequence planning using soft computing methods: A review*, vol. 0, no. 0. 2018.
- [2] I. Ibrahim, Z. Ibrahim, H. Ahmad, M. F. M. Jusof, Z. M. Yusof, S. W. Nawawi, and M. Mubin, "An assembly sequence planning approach with a rule-based multi-state gravitational search algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 79, no. 5–8, pp. 1363–1376, 2015.
- [3] L. Xin, S. Jianzhong, and C. Yujun, "An efficient method of automatic assembly sequence planning for aerospace industry based on genetic algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 90, no. 5–8, pp. 1307–1315, 2017.
- [4] M. A. Abdullah, M. F. F. Ab Rashid, and Z. Ghazalli, "Optimization of Assembly Sequence Planning Using Soft Computing Approaches: A Review," *Archives of Computational Methods in Engineering*, pp. 1–14, 2018.
- [5] Y. Wang and J. H. Liu, "Chaotic particle swarm optimization for assembly sequence planning," *Robot. Comput. Integr. Manuf.*, vol. 26, no. 2, pp. 212–222, 2010.

- [6] J. F. Wang, J. H. Liu, and Y. F. Zhong, "A novel ant colony algorithm for assembly sequence planning," *Int. J. Adv. Manuf. Technol.*, vol. 25, no. 11–12, pp. 1137–1143, 2005.
- [7] A. T. Mathew and C. S. P. Rao, "Implementation of genetic algorithm to optimize the assembly sequence plan based on penalty function," *ARPJ. Eng. Appl. Sci.*, vol. 9, no. 4, pp. 453–456, 2014.
- [8] A. Yasin and N. Puteh, "Product Assembly Sequence Optimization Based on Genetic Algorithm," *Int. J. Computer Sci. Eng.*, vol. 2, no. 9, pp. 3065–3070, 2010.
- [9] Y.-J. Tseng, J.-Y. Chen, and F.-Y. Huang, "A particle swarm optimisation algorithm for multi-plant assembly sequence planning with integrated assembly sequence planning and plant assignment," *Int. J. Prod. Res.*, vol. 48, no. 10, pp. 2765–2791, 2010.
- [10] C. Lu and Z. Yang, "Integrated assembly sequence planning and assembly line balancing with ant colony optimization approach," *Int. J. Adv. Manuf. Technol.*, vol. 83, no. 1, pp. 243–256, 2016.
- [11] M. Li, B. Wu, Y. Hu, C. Jin, and T. Shi, "A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation," *Int. J. Adv. Manuf. Technol.*, vol. 68, no. 1–4, pp. 617–630, 2013.
- [12] A. Mishra and S. Deb, "An intelligent methodology for assembly tools selection and assembly sequence optimisation," *Springer*, 2016.
- [13] C. Lu, Y. S. Wong, and J. Y. H. Fuh, "An enhanced assembly planning approach using a multi-objective genetic algorithm," *J. Eng. Manuf.*, vol. 220, pp. 220–255, 2006.
- [14] H. Lv and C. Lu, "A discrete particle swarm optimization algorithm for assembly sequence planning," *2009 8th Int. Conf. Reliab. Maintainab. Saf.*, no. 2, pp. 1119–1122, 2009.
- [15] C. Hui, L. Yuan, and Z. Kai-Fu, "Efficient method of assembly sequence planning based on GAAA and optimizing by assembly path feedback for complex product," *Int. J. Adv. Manuf. Technol.*, vol. 42, no. 11–12, pp. 1187–1204, 2009.
- [16] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Syst.*, vol. 89, pp. 228–249, 2015.
- [17] B. Bentouati, L. Chaib, and S. Chettih, "Optimal power flow using the moth flame optimizer: A case study of the algerian power system," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 1, no. 3, pp. 431–445, 2016.
- [18] B. Mohanty, "Performance analysis of moth flame optimization algorithm for AGC system," *Int. J. Model. Simul.*, vol. 6203, no. May, pp. 1–15, 2018.
- [19] B. S. Yıldız and A. R. Yıldız, "Moth-flame optimization algorithm to determine optimal machining parameters in manufacturing processes," *Mater. Test.*, vol. 59, no. 5, pp. 425–429, 2017.
- [20] D. Allam, D. A. Yousri, and M. B. Eteiba, "Parameters extraction of the three diode model for the multi-crystalline solar cell/module using Moth-Flame Optimization Algorithm," *Energy Convers. Manag.*, vol. 123, pp. 535–548, 2016.
- [21] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Syst.*, vol. 89, no. July, pp. 228–249, 2015.
- [22] H. Zhao, H. Zhao, and S. Guo, "Using GM (1,1) Optimized by MFO with Rolling Mechanism to Forecast the Electricity Consumption of Inner Mongolia," *Appl. Sci.*, vol. 6, no. 1, p. 20, 2016.