



Landing Gear Disassembly Sequence Planning using Multi-Level Constraint Matrix Ant Colony Algorithm

E.G. Okafor^{1,*}, P.O. Jemitola², O.C. Ubadike³, M.T. Abba⁴, G. Sule⁵ and M.H. Mohammed⁶

^{1,2,3,4,5,6}Faculty of Air Engineering, Air Force Institute of Technology, NAF Base, Kaduna, Nigeria

Abstract

The life cycle of most complex engineering systems is greatly a function of maintenance. Generally, most maintenance operation usually requires the removal of failed part. Disassembly sequence planning is an optimization program that seeks to identify the optimal sequence for the removal of the failed part. Most studies in this area usually, use single constraint matrix while implementing varied complex algorithm to identify the optimal sequence that saves time associated with carrying out maintenance operation. The used of single constraint matrix typically has the drawback of computer higher storage requirement as well as time consumption. To address this problem, this study proposes Multi-Level Constraint Matrix Ant Colony Algorithm (MLCMACA). MLCMACA efficiency was validated using complex aircraft landing gear systems in comparison with genetic algorithms. The result shows MLCMACA superior performance from the perspective of reduced search time and faster tracking of optimal disassembly sequence. Hence is recommended for handling of disassembly sequence planning problems.

Keywords: Algorithm, Constraint matrix, Disassembly sequencing planning, Maintenance, Optimization

1.0 INTRODUCTION

Maintenance plays a major role in equipment life cycle. It simply refers to the work carried out to restore degenerated performance of a system, equipment, or product to its intended normal level [1-4]. Most maintenance often requires the replacement of failed components. Removal and reassembly of these components usually take up a large proportion of time and resources in maintenance task [5]. Disassembly sequence planning problem is a non polynomial NP-hard combinatorial optimization problem, which has attracted many scholars' attention in the past two decades. Studies on disassembly sequence planning (DSP) are the foundation of system disassembly analysis. The main objective of DSP is to find and express the optimal or near optimal sequences of components removal from a system [1, 6]. Generally, disassembly can be broadly classified into three, namely complete, incomplete and selective disassembly [5]. Complete disassembly involves the separation of an entire system into components or subsystems. In contrast, incomplete disassembly only separates some components of the system especially with the intention to recover the original function of systems during its end-of-life. However, the selective disassembly is implemented for the purpose of system maintenance or

upgrade. Compared to the incomplete disassembly, the target component of the selective disassembly is already known [4].

Owing to the importance of the selective disassembly in system maintenance, this paper only focuses on selective disassembly. Normally, matrices are introduced to represent constraints, precedence relationships, interferences and structure information of a product [8]. A lot of different matrices have been proposed to express product information such as part constraints, fasteners and interferences [1, 5]. These matrices can then be used to search for feasible sequences. With the view of getting feasible and optimal disassembly sequence more effectively, techniques such as graph theory, Petri nets, linear programming, as well as meta-heuristics techniques such as, genetic algorithm (GA), ant colony algorithms (ACA) are widely used in this field.

To illustrate, Zhang and Kuo used a graph based disassembly tree for the recycling of personal computers [9]. Lambert presented mixed integer programming and linear programming based method separately to obtain the optimal disassembly sequence according to graphic description of disassembly object [10]. Wu and Zuo presented a hybrid graph based genetic/simulated annealing algorithm with binary-tree to solve the optimization problem of disassembly sequence planning [11]. Smith et al. [9] developed a rule-based recursive selective disassembly method for finding a near-optimal disassembly sequence for products. They introduced four matrices and five disassembly rules. The four matrices are: disassembly matrix for fasteners (DF), disassembly matrix for components (DC), motion constraints for

*Corresponding author (Tel: +234 (0) 816 660 5144)

Email addresses: eg.okafor@gmail.com (E.G.

Okafor), pojemitola@yahoo.co.uk (P.O. Jemitola), diketronics@yahoo.com (O.C. Ubadike), m.tahir.abba@gmail.com (M.T. Abba), gowon.sule@yahoo.com (G. Sule), mdhabeebgololo@gmail.com (M.H. Muhammad)

fasteners (MF) and motion constraints for components (MC). The five rules define the disassembly order or priorities among components. They then introduced a cost function in the method to find the near optimal disassembly sequence with the lowest cost. By using this method they solved selective disassembly sequence planning problems with both single-target component and multiple-target components. Lu et al [4] implemented Swarm Particle algorithm to solved DSP of a landing gear.

Heuristic algorithms usually use functionality, performance, manufacturability, reliability, eco-compatibility etc., as criteria to select the near optimum disassembly sequence [7]. Comparing with the graph-based searching method, heuristic algorithms are much more efficient. Genetic algorithms (GA) and ant colony searching methods belong to the large class of Heuristic algorithms [7, 13, 14]. GA mimics the process of natural evolution processes [14]. Using GA, a number of strings are generated. The fittest ones are used for further analysis. Detailed description of GA can be found in [15-17]. Genetic algorithms can provide the near optimum disassembly sequence in a short time. This method can consider precedence relationships and additional constraints in the product structure [13, 18]. Xie et al., [19] developed a disassembly sequence planning method named simulated annealing and genetic algorithm (SAGA) which combines the benefits of simulated annealing (SA) and genetic algorithm (GA). Parsa and Saadat [20] investigated automated disassembly using the genetic algorithm and proposed a model for robotic disassembly sequence optimization. Other optimization methods such as particle swarm optimization algorithm have been used to solve multi-objective optimization problems [21].

Ant colony optimization algorithm (ACOA) was initially proposed by Marco Dorigo [14] in his thesis. The method was designed to mimic the behaviour of ants seeking a path between their colony and a source of food. ACOA has been shown to be efficient in solving computational problems. Based on ACOA, Wang et al. [22] solved the disassembly problem of an automotive engine, which consists of 50 parts after the simplification. Disassembly Feasibility Information Graph (DFIG) was defined in the paper. Using the DFIG, the problem of disassembly path planning was transformed into the problem of searching near optimal sequence in their directed, weighted graph. In conclusion, ant colony algorithm is an efficient search method, which can be used to solve practical problems and products with lots of elements.

Although so many techniques have been proposed, few of them are suitable for maintenance disassembly of large equipment such as aircraft and weapon systems. For instance, numerous disassembly sequence planning methods focus on the field of recycle design [23]. Recycle disassembly is a multi-component

disassembly conducted to remove all reusable components from equipment. In maintenance disassembly sequence planning, minimization of maintenance time and cost is usually taken as objective. Generally, past studies have concentrated on the use of single constraint matrix to represent system or product of interest. This is found to waste large computer storage space as well as increase search time [7]. To address this issue of large storage space and search time associated with single constraint matrix, this study adopts a technique which replaces the single matrix with a multi level constraint matrices, then implement ant colony algorithm to search for the optimal sequence. The approach works on the principle of hopping from a higher level matrix (system) to lower level matrix (sub assembly) until the target part is removed. The method consists of three main parts: representation of product constraint matrixes, searching, and evaluating of disassembly sequences. Simulation of the near optimal result is built afterwards. The rest of the paper is organized as follows. In section 2 constraint matrix is described. A brief description of ant colony optimization operational principles is given in section 3. In section 4 the study research method is presented with emphasis on the Multi-level constraints matrices, problem formulation and Multi-level constraints matrix ant colony algorithm. Section 5 presents the study parameter setting. In section 6 the study results are presented and discussed, while section 7 concludes the study.

1.1 Constraint Matrix (CM)

Constraint matrices are very popular product representation method in terms of product structure information, parts constraints and precedence. There are numerous kinds of constraint matrices. Some of them are used to represent relationships between every two parts, while others simply consider the contact parts. A lot more consider directions. Irrespective of which matrix is to be implemented, there is always a rule to follow. Also, it is more convenient for the computer to search for sequence using matrices in comparison to graphs. CM representation provides a possibility for dealing with DSP problems with numerous parts. A general formation of a constraint matrix is represented as shown in Fig. 1.

$$\begin{bmatrix} b_{11} & b_{12} & b_{11} & \dots & b_{1m} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2m} \\ b_{31} & b_{32} & b_{33} & \dots & b_{3m} \\ \dots & \dots & \dots & \dots & \dots \\ b_{m1} & b_{m2} & b_{m3} & \dots & b_{mm} \end{bmatrix}$$

Figure 1 Constraint matrix of subassembly

The matrix has an initial size of $m \times m$, where m stands for the number of subassemblies or components in the system or subsystem respectively. Elements of the matrix b_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, m; i \neq j$) is either

0, 1 or -1. They stand for the restriction relationships between subassemblies. The “1” indicates that one element will stop another one from moving in the positive direction. In contrast, the “-1” illustrates that the element will stop another one from moving in the negative direction. In this research, only relationships between contacted subassemblies will be concerned, otherwise b_{ij} state will be set as “0”, where there are no direct restrictions between these subassemblies. The principal diagonal elements of the matrix stand for the time required to disassemble a subassembly. For example, $b_{ij}=70$ ($i = j = m$) represents that 70 seconds is required to disassemble the subassembly or component m . The precise time required for disassembling a subassembly can be calculated using criteria described in ref [24], and introduced in Section 2.3.

Chung and Peng developed a disassembly system based on the constraint matrix representation, which improved the automatic level of sequence planning problem [2]. Wu and Peng also solved the product maintainability evaluation problem based on an improved constraint matrix, which contributed to the efficiency of product life-cycle operations and the reduction of operation cost [25]. Although constraint matrix representations have many merits, they equally have demerits. Usually, the matrix records relationships between every two parts even though there is no constraint between the two parts. This implies that, some recorded data in the matrix do not have to be considered, which clearly reveals the problem of storage space wastage as well as increase search time. This study seeks to address the problem of computer storage space wastage and increase search time peculiar with single DSP based on

single constraint matrix using multi-level constraint matrices ant colony algorithm (MLCMACA).

1.2 Ant Colony Algorithm

In the realistic terms, ants travel randomly in search for food. Upon finding food source it lays down pheromone trail to facilitate their way back to the colony. The positive feedback eventually leads all ants to following that path that leads to food spot [26]. Subsequently random movement will stop [22]. The inspiration of the ant colony operation algorithm is to mimic ants' behavior. A detail description of this algorithm can be found in [14]. In disassembly sequence planning problem, if a component cannot be disassembled, along a path, then this path is unacceptable, and there is no need to consider the components afterwards. Figure 2 shows the operational principles of the ant colony algorithm [27]. From figure 2 paths 1-2-5-3 and 4-3-2 cannot be used to disassemble component 3 and 2 respectively. A tour of ants can be used to represent an acceptable component disassembly solution. If an ant has passed N-1 possible operation nodes of its route, it moves to the next node where the operation possibility is false. That means the route is unacceptable, and the following nodes will not be calculated. By applying this principle for disassembly sequence planning, the computational complexity can be reduced when compared with genetic algorithm. Also, the efficiency of the searching process is also improved. Moreover, if the target part is already disassembled in the previous operation, the searching stops and no other components need to be considered in future operations. Once a possible path is found, criteria will be used to evaluate the quality of the path for a near optimum sequence.

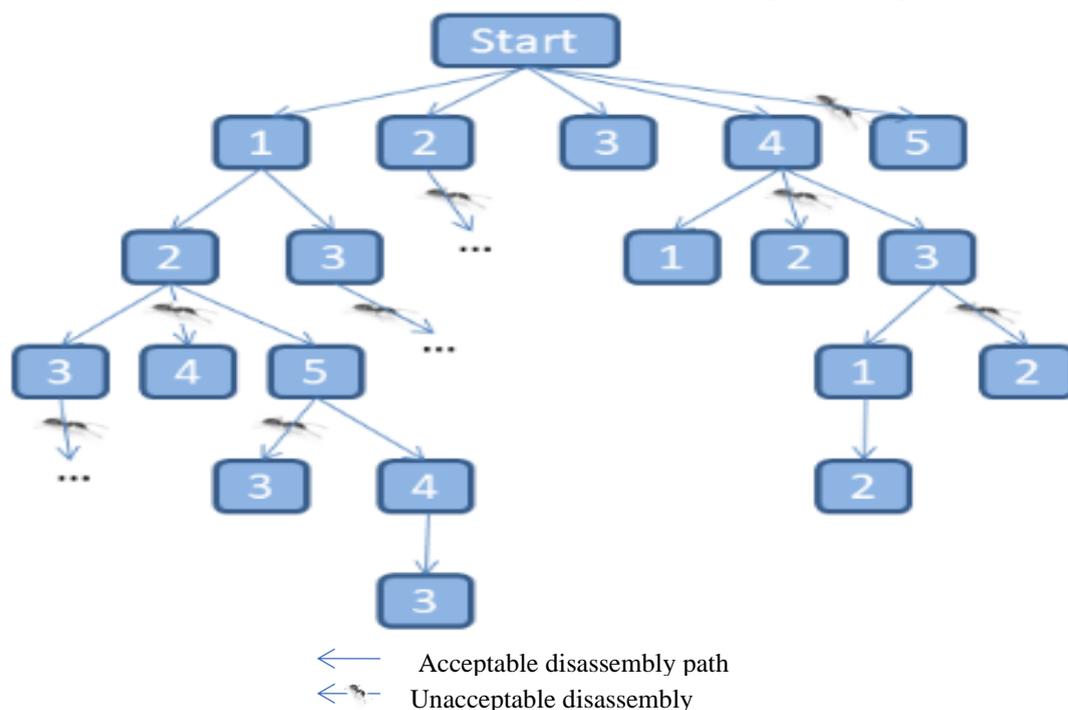


Figure 2: Operational principle of disassembly sequence based on ant colony algorithm

2.0 METHOD

2.1 Multi-Level Constraint Matrices

In a disassembly process, a system is initially divided into a number of first-level subassemblies, and the system structure is represented in a constraint matrix based on the relationship between the first-level subassemblies and the predominant constraints rule adopted. In this work the design information was obtained from the nose landing system for general aviation type aircraft, and each level of landing gear subassemblies was identified and labeled with a number. Since most components can only be removed from one direction, only one-degree motion along axial direction of its local coordinate system is considered. Figure 3 shows the model landing gear system considered. First-level subassemblies contain three second-level subassemblies (see Figure 4a), and each of the first-level subassemblies is represented by one constraint matrix. These matrices are the second level matrices (see Figure 4b). The generating method and information of these second level constraint matrices are almost the same as the first level constraint matrix. The difference is that the second level matrices represent relationships between components instead of subassemblies. The multi-level matrices representation of a system is efficient in solving disassembly sequence planning problems. For instance, smaller memory is required to save multi-layer matrices than one level matrix when dealing with complex problems.

To illustrate, in Figure 3, the landing gear system considered consists of 18 components, and these 18 components form three subassemblies 1,2,3 and represented using *Shock Strut Assembly (SSA)*, *Torquelink Assembly (TLA)*, *Wheel Assembly (WA)* respectively. SSA contains eight

components, while *TLA* and *WA* are made up of 4 and 6 components respectively. Using one level matrix, $18 \times 18 = 324$ elements need to be recorded.

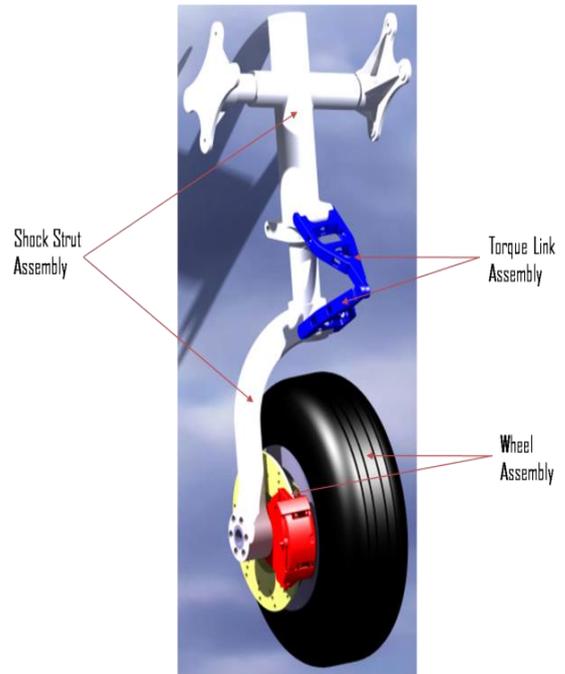


Figure 3: Landing gear system considered

However, by using the multi-layer matrices representation method, this landing gear system can be represented by a set of matrices, (one 8×8 matrix, one 4×4 matrix, and one 6×6 matrix). Then, $(8 \times 8) + (4 \times 4) + (6 \times 6) = 116$ elements will be recorded. In this study, constraint matrix was generated in agreement with the procedure described in section 2.1. The multi-layer matrices and the single-level matrix representation of the landing gear are depicted in Figures 4 and 5, respectively.

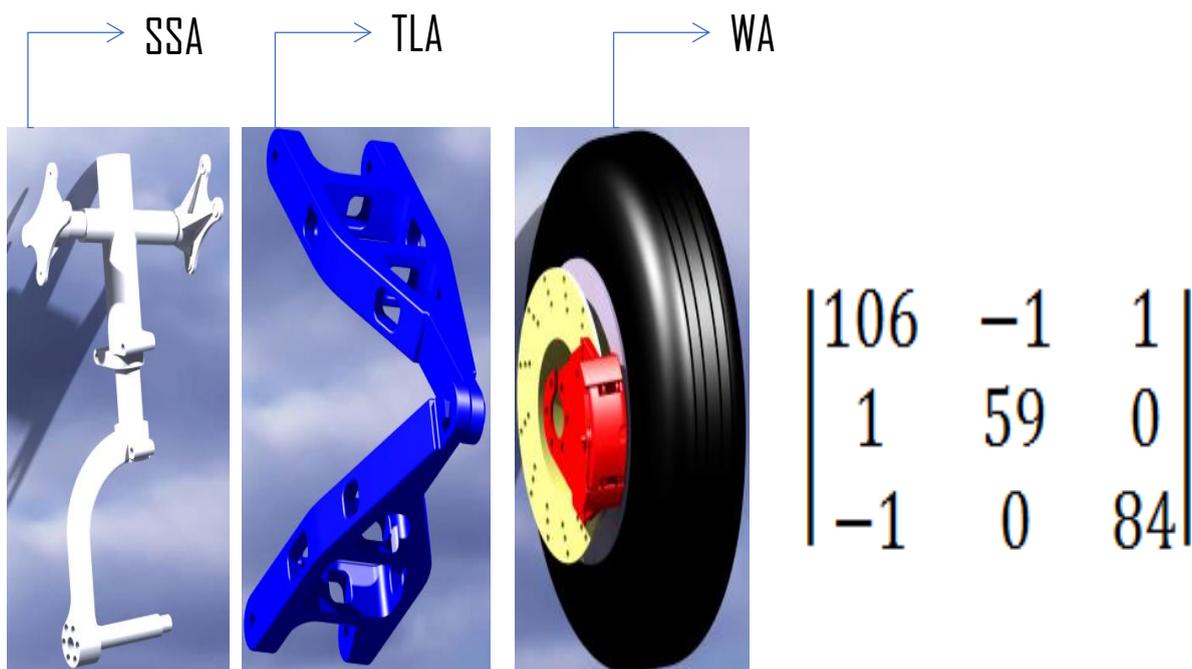


Figure 4a: Multi-Level Matrices Representation of the landing gear – First Level Matrix

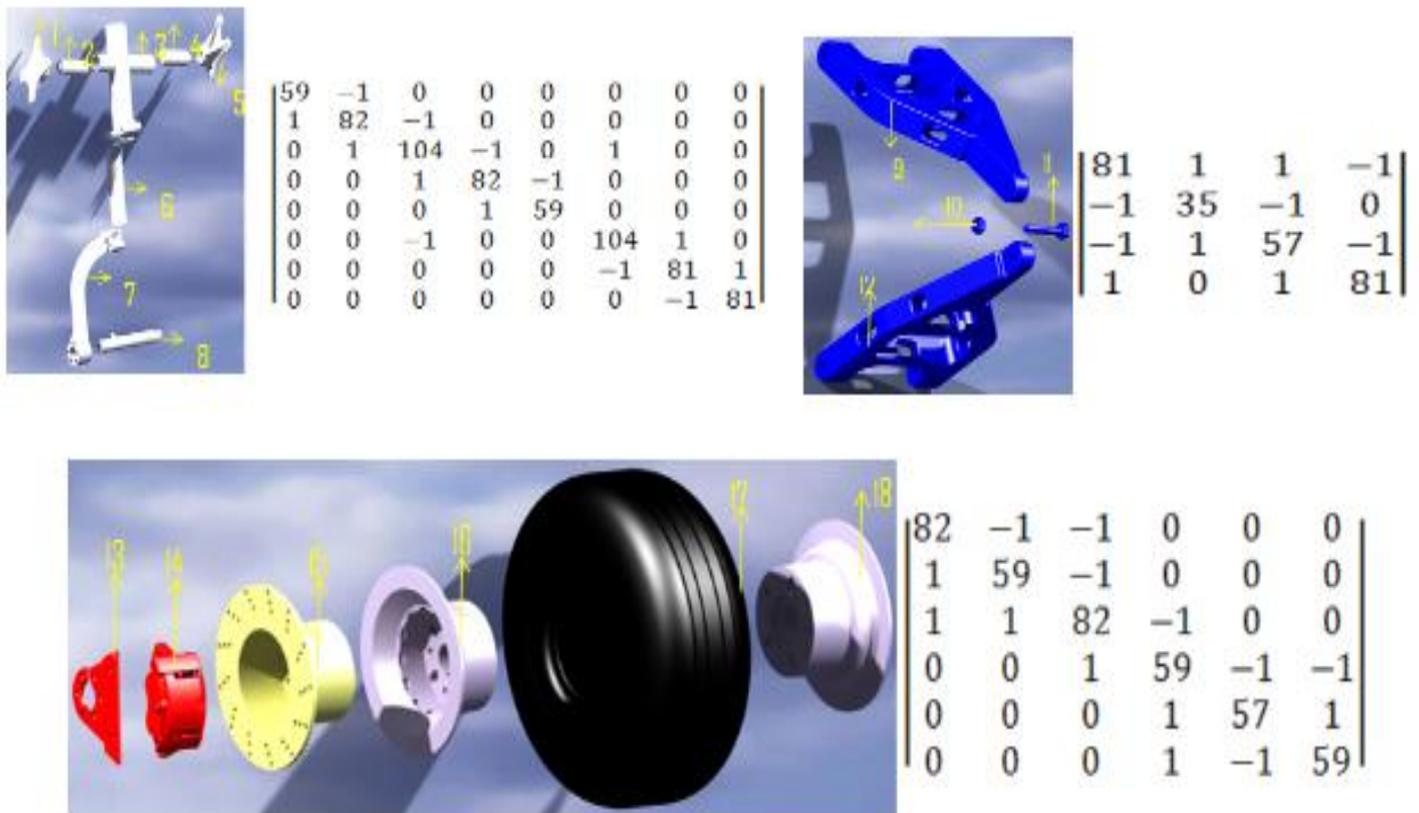


Figure 4b: Multi-Layer Matrices Representation of the landing gear – Second Level Matrix

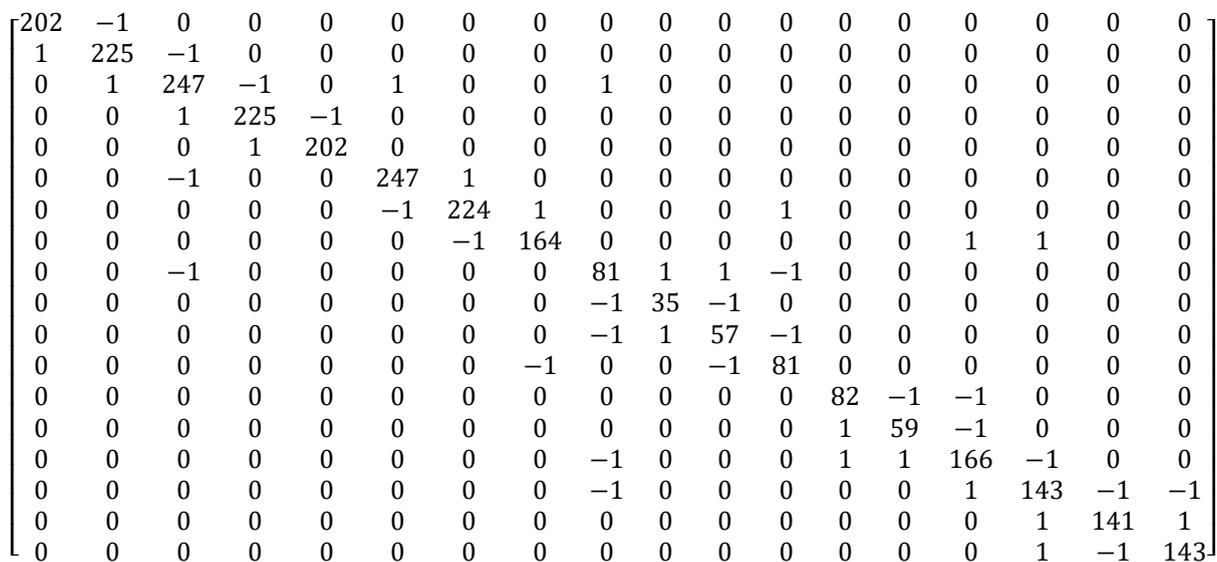


Figure 5: One Level Matrix Representation of the landing gear

2.2 System part disassembly time estimation criteria

The time required to disassemble a component varies from one to another depending on the part location and complexity. Table 1 show the relationship between component complexity and additional time required to disassembly the components [24]. In this work a component disassembly time was computed as a function of basic and additional time. The basic time was set, while the additional time was established in accord with the

criteria (geometry and process) described in Table 1.

2.3 Problem formulation

Based on the above criteria, the total disassembly time required to remove the target part can be calculated for all visible sequences. In other to select the optimal sequence the objective function implemented in this work is given in Eq. (1).

$$\min f(x) = [T_t, N_c]$$

Subject to

$$\begin{aligned} j &\leq 6 \\ N_c &\leq 10 \end{aligned} \tag{1}$$

T_t is computed using Eq. (2).

$$T_t = \sum T_e + \sum_{j=1}^6 T_{mj} \tag{2}$$

where T_e stands for the basic time required for the disassembling of a regular part. A regular part in this research is defined as a part with regular size (larger than 64 cm³ and smaller than 6.4x10⁴ cm³), regular weight (less than 20kg) and normal shape, which will not bring difficulty in a disassembly operation. In this study, $T_e=15s$, while j is the criteria number listed in Table 1, and i is the number of the removed part. T_{mj} represent the additional time required for different criteria (see Table 1).

Table 1: Part Disassembly Complexity [24]

Criterion	j	Level of complexity	Additional time required (second)
Geometry complexity	1	Part size:	
		i.Regular size	0
		ii.Small size	23
		iii.Large/heavy	25
	2	iv.Super large/heavy subassembly	86
		Handling difficulty:	
	3	v.Tool or fixture required	20
		vi.Difficulty	18
		Features may cause jam and tangle:	
		vii.Yes	12
Process complexity	4	viii.No	0
		Fastening type:	
		ix.Multi-pieces (nuts and bolts)	24
	5	x.Screws and nails	24
		xi.Rivets, staples and adhesive	20
		Enough space for disassembly operations	
	6	xii.Yes	0
		xiii.No	2s
		Interference with other parts	
		xiv.Yes	22
	xv.No	0	

If part l satisfy the criteria j , then the corresponding additional time T_{mj} will be added to the total disassembly time of this component m . Otherwise, the additional time will not be included. If a subassembly contains several components, the total removal time of these components can be calculated by adding all removing time spent for each of the different components in the sequence.

The total number of components disassembled for a sequence is N_c . When both the least disassembly time and minimum number of components cannot be met, the least disassembly time associated with a sequence is considered as more important than the total number of components disassembled, and a near optimum sequence will be applied.

2.4 Multi level constraint matrices ant colony algorithm (MLCMACA)

The essential principles of the ACOA as well as its advantages have been presented in section 1.2. In this study **MLCMACA** is proposed. **MLCMACA basically** hops from one matrix level to another in search for the target part. A flowchart that depicts the MLCMACA is given in Figure 6, while the algorithm is described in section 2.4.1.

2.4.1. The Algorithm

Figure 7 represents the flowchart of the MLCMACA. The algorithms associated with the proposed MLCMACA are:

Step 1: Set the number of iteration N

- Step 2:** Set the target part number
- Step 3:** Set the number levels
- Step 4:** Input all matrixes associated with each level
- Step 5:** Is the target part in the sub assembly of first level matrix?
- Step 6:** If Yes to step 5, implement the following:
- i. Randomly get one element from i^{th} line of the first level matrix.
 - ii. Does the i^{th} line of the matrix contain both 1 and -1? If yes go to step 6a. if No, Remove the element which equal to "i" from the matrix and put it in a trace (1 by z matrix)
- Step 7:** If No to step 5 go back to step 5
- Step 8:** Move to the lower level matrix and of the selected sub assembly from the preceding level.
- Step 9:** Is the target part lowest level matrix reached? "Yes" go to Step 11, "No" go to step
- Step 10:** Insert the element after the last element in the trace
- Step 11:** Compute the total disassembly time using Eq. 2.
- Step 12:** Is the number of iteration 1?
If "Yes" then the disassembly time (T_s), total number of element in the trace (N_m) and sequence (trace).
If "No" is $T_s < \text{new disassembly?}$ "Yes" then set new disassembly time = T_s , new total number of element in the trace = N_m , and the new sequence is the trace. If "No" T_s , N_m and the trace is unchanged
- Step 13:** Is the number of Iteration = n ? "No" go to step 5. "Yes" display the Optimal time as T_s , and the optimal sequence as "trace".

Once a matrix is generated, the disassembly ability of each component is checked. If the component cannot be disassembled, sequences starting with this component will not be searched any longer, and other

components will be checked continuously. Otherwise, the component disassembled will be recorded and compared to the target part. If the component is not the target part, the matrix will be simplified and the next level searching starts. If the component is the target part, the disassembly sequence will be recorded and total time calculated. This disassembly total time is used to compare to the minimum time already established. If the time is lower than the minimum time already established, the disassembly sequence and its corresponding time will be used to replace the optimized sequence and minimum time. If the disassembly time equals to the minimum time, the number of components removed will be considered. Sequence with fewer components movements will be saved as the optimized sequence and minimum cost. Otherwise, if the disassembly time is larger than the minimum time, the optimized sequence and minimum time is unchanged, and the system continues searching other sequences. Following these processes, an optimized sequence with the least disassembly time and least number of disassembled components will be finally generated.

2.5 Parameter Settings and Comparison

The example of a nose landing gear used to illustrate the efficiency of the MLCMACA comprises of two levels matrix. A detailed description of the landing gear system based on the concept of multi level matrices is given in section 2.1. Two hundred and fifty (250) search iterations were used to validate the proposed technique. Component "17" was set as the target part. The efficiency of MLCMACA was compared with genetic algorithm. The implementation of GA was carried in agreement with the procedure described in [28]. In line with [28], population size of 20, crossover rate of .75, mutation rate of 0.05 and 250 generations were used for the GA implementation. All programs were executed on a computer with specifications as follows; RAM 4GB, 64-bits operating system and AMD Athlon II P320 dual core processor.

3.0 RESULTS AND DISCUSSION

The result of the MLCMACA and genetic algorithm for the two level matrices used are shown in figures 7 and 8 respectively. From figure 7 it is shown that a total of 115 second was required to successfully disassemble component "17" using MLCMACA. Furthermore the optimal sequence was found to be WA, 18, 17. Table 2 compares genetic algorithm and MLCMACA over the search time, number of iteration which led to optimal sequence and optimal sequence. The MLCMACA was found to have a faster search time, in comparison to the GA technique. Similarly, the optimal was obtained after 3 iterations for MLCMACA and 6 iterations for GA. This result suggests that MLCMACA is more preferable for disassembly problems when compared to GA.

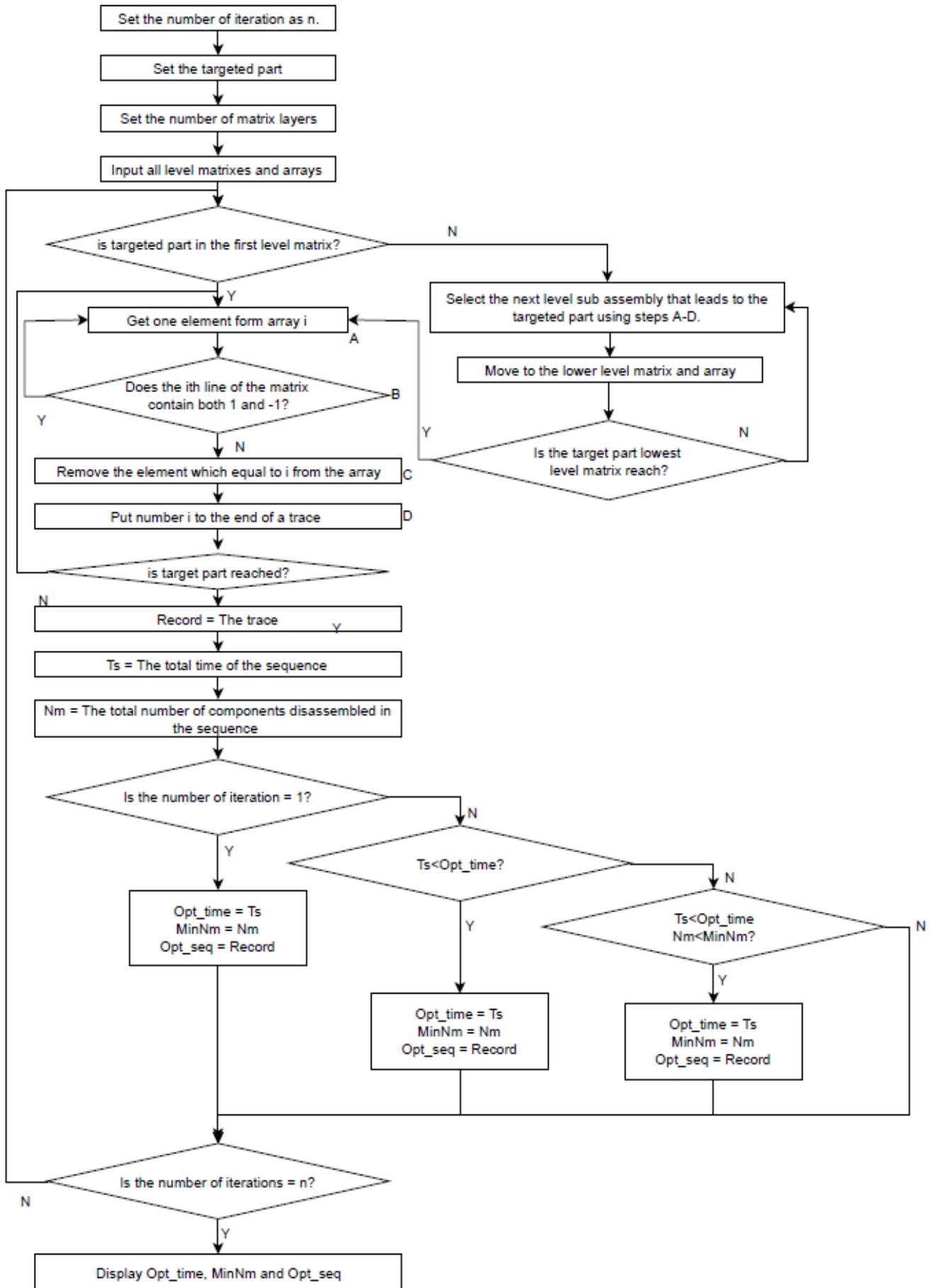


Figure 6: Flowchart for MLCMACA

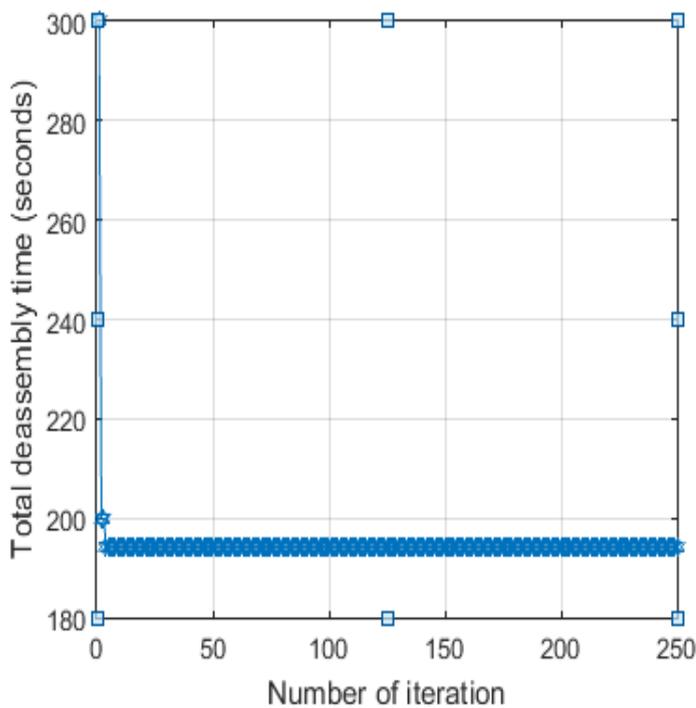


Figure 7: Plot of MLCMACA total disassembly time against number of iterations

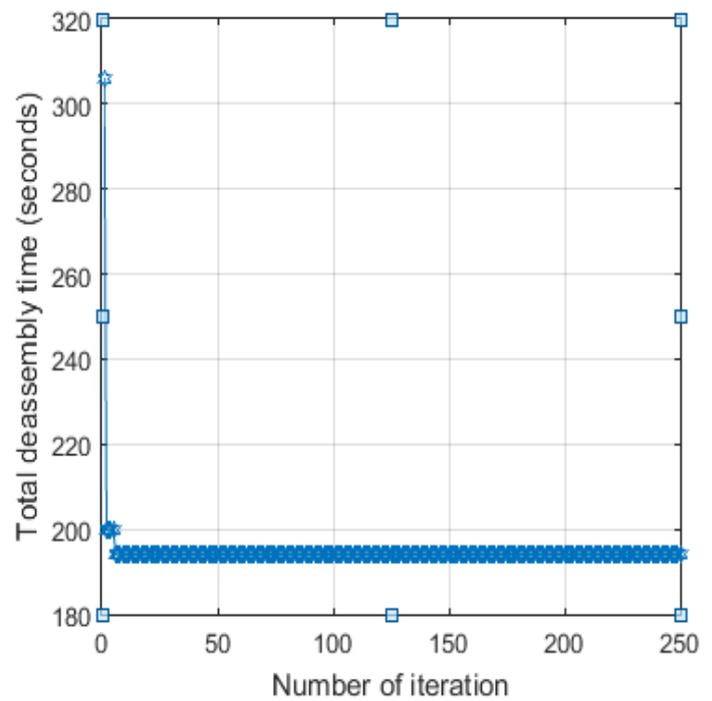


Figure 8: Plot of GA total disassembly time against number of iterations

Table 2: Comparison of different methods

Methods	Target part	Total search time (seconds)	Optimal sequence	Generation of identification of optimal sequence
Two level genetic algorithm	Component 17	140	WA, 18, 17	6
MLCMACA		115	WA, 18, 17	3

6. CONCLUSION

Landing gear disassembly sequence planning was carried out using MLCMACA, in comparison to GA. The multi-level matrices representation has reduced the search size greatly. The removal of unnecessary data associated with MLCMACA, avoided unnecessary searches. The ant colony operation algorithm can further reduce the search size, which helps to reduce the searching complexity. The combination of the multi-level constraint matrices representation method and ant colony searching method improved the efficiency of disassembly sequence planning, especially for the product with many parts. The evaluation criteria guaranteed a near optimal sequence selected, the program developed helps to improve the automatic level.

ACKNOWLEDGEMENT

This work was supported by Tertiary Education Trust fund under the Institutional Based Research (IBR) grant 2019.

REFERENCES

[1] Soran, P., and Mozafar S. "Intelligent selective disassembly planning based on disassemblability characteristics of product components". *The International Journal of Advanced Manufacturing Technology*, (2019), <https://doi.org/10.1007/s00170-019-03857-1>

[2] Smith, S. and Hung, P-Y. "A novel selective parallel disassembly planning method for green design." *Journal of engineering design* 26(10-12), (2015), 283-301.

[3] Ziqiang, Z., Guohong, D., Xiangyan, Z., Chaobin, H. and Yongjian, Z. "Research of partial destructive based selective disassembly sequence planning." *The Open Mechanical Engineering Journal* 9(1), (2015).

[4] Lu Z., Sun Y-C, Okafor E.G., and Wu, H.Q. "Disassembly sequence planning for maintenance based on metaheuristic method". *Aircraft Engineering and Aerospace Technology: An International Journal*, 83(3), (2011), 138-145.

- [5] Chung, C., and Peng, Q. "Evolutionary sequence planning for selective disassembly in demanufacturing". *International Journal of Computer Integrated Manufacturing*, 19(3), (2006), 278-286.
- [6] Yongtao, L. "Disassembly Sequence Planning for Product Maintenance". *MSc Thesis, University of Manitoba, Canada*. (2014)
- [7] Giudice, F. "Disassembly depth distribution for ease of service: a rule-based approach". *Journal of Engineering Design*, 21(4), (2010), 375-411
- [8] Tang, Y., Zhou, M., Zussman, E., and Caudill, R. "Disassembly modeling, planning and application: a review". *IEEE International Conference on Robotics and Automation*, 3, (2000), 2197-2202.
- [9] Zhang, H.C. and Kuo, T.C. "A graph-based disassembly sequence planning for EOL product recycling". *Proceeding of the 21st IEEE/Cpmt International Electronics Manufacturing Technology Symposium in Austin, USA*, (1997), 140-151.
- [10] Lambert, A.J.D. "Optimum disassembly sequence with sequence-dependent disassembly costs". *Proceeding of the 2003 IEEE International Symposium on Assembly and Task Planning in Besan Gon, France* (2003), 151-156.
- [11] Wu, H and Zuo, H.F. "Using genetic/simulated annealing algorithm to solve disassembly sequence planning". *Journal of systems Engineering and Electronics*, 20(4), (2009), 906-912.
- [12] Smith, S. S., and Chen, W. H. "Rule-based recursive selective disassembly sequence planning for green design". *Advanced Engineering Informatics*, 25(1), (2011), 77-87.
- [13] Mohanta, J.C., Parhi, D. R., and Patel, S. K. "Path planning strategy for autonomous mobile robot navigation using Petri-GA optimization". *Computers & Electrical Engineering*, 37(6), (2011), 1058-1070.
- [14] Dorigo, M. "Optimization, learning and natural algorithms". *Ph. D. Thesis, Politecnico di Milano, Italy*. 1992.
- [15] Okafor, E.G., Uhuegho, K.O., Sun, Y-C. "Multi-Objective Optimization of an Aircraft Position Indicator and Actuation System (PIAS) using GPSIA+DS". *Recent Patent on Engineering*, 10, (2016), 51-68.
- [16] Okafor E.G., and Sun Y-C. "Multi-objective optimization of a series-parallel system using GPSIA" *Journal of Reliability Engineering & System Safety*, 103, (2012), 61-71.
- [17] Okafor E.G., Uhuegho, O.K., and Sun, Y-C. "Multi-objective optimization of a complex system using GPSIA+DS". *Research Journal of Applied Science and Engineering Technology*, (2013), 4621-4629
- [18] Tseng, H.E., Chang, C.C., Lee, S.C. and Huang, Y.M. "A block-based genetic algorithm for disassembly sequence planning." *Expert Systems with Applications*, 96, (2018), 492-505.
- [19] Xie, Y. F., Huang, M. F., Zhong, Y., and Kuang, B. "Disassembly sequence planning based on the simulated annealing and genetic algorithm". *Design and calculate*, (2007), 103-105.
- [20] Parsa S. and Saadat M. "Intelligent planning using genetic algorithm for automated disassembly". *Proceeding of 16th International Conference on Manufacturing Research ICMR2018*, (2018). <https://doi.org/10.3233/978-1-61499-902-7-189>.
- [21] Tao, Y., Meng, K., Lou, P., Peng, X. and Qian, X., Wang, L., Cai, R., Jing, L., and Zhang, H. "Object-guided ant colony optimization algorithm with enhanced memory for traveling salesman problem". *Research Journal of Applied Sciences, Engineering and Technology*, (2012), 3999-4006.
- [22] Fernando, T. and Santiago, P. "Intelligent disassembly in the demanufacturing process". *International Journal of Advanced Manufacturing Technology*, 30(5/6), (2006), 479-80.
- [23] Li, W., Peng, Q., and Xing, M. "Bioreactor improvement based on design for assembly in virtual environments". *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (2011), 733-742.
- [24] Wu, N., and Peng, Q. "Maintainability evaluation based on the product disassembly analysis". *ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, (2009), 101- 109.
- [25] Eldos, T., Kanan, A., and Aljumah, A. "Solving the printed circuit board drilling problem by ant colony optimization algorithm". *World Congress on Engineering and Computer Science*, (2013), 584-588.
- [26] Malik, S. "Performance comparison between ant algorithm and modified ant algorithm". *International Journal of Advanced Computer Science and Applications*, (2010), 42-46.
- [27] Okafor, E.G., Jemitola, P.O., Manshop, C. and Uhuegho KO. "A Study of Airline Fuel Planning Optimization Using R-GA". *Journal of Engineering Research in Africa*, 46, (2019), 186-197.