An effective teaching learning based optimization for flexible job shop scheduling

Raviteja Buddala
Ph.D. Research Scholar, Department of Mechanical Engineering
National Institute of Technology
Rourkela-769008, Odisha, India
Email: raviteja317@gmail.com

S. S. Mahapatra
Professor, Department of Mechanical Engineering
National Institute of Technology
Rourkela-769008, Odisha, India
Email: mahapatrass2003@gmail.com

Abstract— A flexible job shop scheduling problem (FJSSP), a prolongation of former job shop scheduling problem (JSSP), is well known as a NP-hard (Non-deterministic polynomial time) problem. The FJSSP contains the difficulties of classical JSSP with an additional complexity of allotment of operations to the machines. A FJSSP has two sub problems. They are (i) routing sub problem i.e. allotting of each operation to a machine from a given set of machines capable of doing it and (ii) sequencing sub problem i.e. arranging of allotted operations on each machine. It is not always possible to get an optimal solution to such problems in a reasonable computation time owing to large computational complexity involved. Therefore, a variety of meta-heuristic approaches have been used to obtain near optimal solutions in a reasonable computation time. Teaching learning based optimization (TLBO), which needs minimum number of parameters to be tuned in comparison to other meta-heuristic approaches, can be an efficient algorithm. Like many meta-heuristics, TLBO may get trapped at the local optimum and present difficulty in reaching the best solution. In this paper, this limitation is addressed efficiently by embedding TLBO with a new effective local search technique. Real number encoding system is used and the FJSSP is solved based on integrated approach. Tests are carried out on famous bench mark problems with minimization of makespan as the objective. Results obtained from modified TLBO have been compared with other algorithms from literature.

Keywords—flexible job shop scheduling; local search; makespan; meta-heuristic; TLBO.

I.INTRODUCTION

Scheduling is a crucial decision making aspect in both manufacturing and service sectors to improve the customer satisfaction and organizational effectiveness. Scheduling deals with the allotment of tasks on capable machines in such a way that some performance criteria like lateness, flowtime, tardiness and makespan can be minimized. In the present competitive world, effective scheduling has become absolute necessary for survival in the market. Manufacturing industries must meet the due date committed to customers to safeguard the goodwill of the company. The industries must schedule the activities in such a way that available resources could be utilized in an effective way. FJSSP, an extension of classical JSSP, is a well-known NP-hard problem [1] where operations are processed on a given set of machines available. The FJSSP contains the difficulties of classical JSSP with an additional complexity of allotment of operations to the machines. A FJSSP has two sub problems. They are (i) routing sub problem i.e. allotting of each operation to a machine from a set of machines capable of doing it and (ii) sequencing sub problem i.e. arranging of allotted operations on each machine. The FJSSP can be solved in two different ways. They are *(1) Hierarchical approach,* in which routing problem and sequencing problem are solved one after the other. The problem is divided into two parts to increase the ease of solving the problem. Brandimarte [2] has applied hierarchical approach for the first time using dispatching rules for routing sub problem and Tabu Search (TS) for sequencing sub problem. Kacem et al. [3-4] have proposed a Genetic Algorithm (GA), incorporated by the approach of localization (AL) approach for solving multi-objective and mono-objective FJSSP in the hierarchical approach. *(2) Integrated approach*, is one in which routing and sequencing sub problems are solved simultaneously. Generally integrated approaches are harder to solve but produce better solutions than the hierarchical approaches. Chen et al. [5] and Jia et al. [6] used integrated approach using GA, Bagheri et al. [7] used an integrated approach using Artificial Immune Algorithm (AIA), Yuan et al. [8] used an integrated approach using Hybrid Harmony Search (HHS), Singh et al. [9] and Singh & Mahapatra [10] used an integrated approach with different forms of Particle Swarm Optimization (PSO).

II. LITERATURE REVIEW

Two functions have been proposed by Mastrolilli and Gambardella [11] to generate neighborhood solutions so that they can be employed in meta-heuristics to work out the FJSSP. Xia and Wu [12] have proposed a hierarchical approach with PSO and Simulated Annealing (SA) such that PSO solves routing sub problem and SA solves the sequencing sub problem. A Hybrid hierarchical approach of Ant Colony Optimization (HACO) and Tabu Search (TS) have been used by Liouane et al. [13] where the

978-1-4673-9939-5/16/$31.00 ©2016 IEEE

ACO is employed to solve the routing sub problem and TS is used to work out the sequencing sub problem. To solve multi-objective FJSSP, Xing et al. [14] proposed a simulation model which is a general frame work that can be used with many algorithms. A new local search algorithm has been proposed by Xing et al. [15] to solve both the partial flexible job shop scheduling problems (P-FJSSPs) and total flexible job shop scheduling problems (T-FJSSPs). In T-FJSSP an operation can be done on any machine from the machines available where as in P-FJSSP one or more operations can be done only on selected available machines. With an efficient neighborhood structure for FJSSP, Li et al. [16] have proposed a Hybrid Tabu Search Algorithm (HTSA) where TS solves the routing sub problem and variable neighborhood search (VNS) is used to solve sequencing sub problem. An Artificial Bee Colony algorithm (ABC) for FJSSP is proposed by Wang et al. [17] where they have imported the cross over and mutation techniques of genetic algorithm and solved the routing sub problem then the sequencing sub problem is solved with a critical path based local search strategy. A new biogeography based optimization algorithm (BBO) is proposed by Habib et al. [18] to solve the FJSSP where it is claimed that BBO is better than GA in finding the solutions. Gao et al. [19] have proposed a Discrete Harmony Search (DHS) algorithm to solve the FJSSP with multiple objectives. A Multi Objective PSO (MOPSO) is proposed by Singh et al. [9] for solving FJSSP to find approximate pareto-optimal solutions. Here they have used an integrated approach to solve the problem. Sun et al. [20] have proposed a Hybrid Bayesian optimization based algorithm (HBOA) to solve the FJSSP which proved that HBOA is better than the original BOA. Chang et al. [21] have proposed a hybrid Taguchi-Genetic Algorithm to solve the FJSSP with makespan as objective. This approach gives better or equal results compared to previous literature except three problems where Zhang et al. [22] is superior for these problems. In the current work, the problem mapping and solution generation mechanism is taken from Singh and Mahapatra [23].

III. TEACHING LEARNING BASED OPTIMIZATION

TLBO is proposed recently by Rao et al. [24-26] from the general teaching learning process, considering the output of the class with an effect of influence of teacher. Students and teacher are the two crucial entities of the algorithm and explains the two primary ways of learning, via teacher (also called teacher phase) and discussing with the co-students (also called student phase). The TLBO is a population based algorithm and a group of students constitute a population. The best solution in the total population is taken as teacher. The execution of this algorithm is split into two components, they are "teacher phase" and "student phase" as explained later.

A. Teacher Phase

A teacher increases the average value of the class from any value M_x to any other better value M_y depending on teachers capacity. In any iteration i, let T_i be the teacher and M_j be the mean of the class. Such that teacher T_i tries to raise the mean of the class given by

$$\text{DifferenceMean} = r_i(T-T_f*M_j) \tag{1}$$

Here r_i is a random number. T_f is the only parameter in this algorithm known as teaching factor. The T_f value can either be 1 or 2, is a probabilistic value that is taken randomly with same probability as follows

$$T_f = \text{round}(1+\text{rand}) \tag{2}$$

Depending on these two equations, the new solutions are updated as follows where Y denotes student of the class.

$$Y_{newi} = Y_{oldi} + \text{DifferenceMean} \tag{3}$$

Accept $Y_{new\,i}$ if it generates a better makespan value.

B. Student Phase

The next part of the TLBO where students improve their knowledge by discussing among themselves. A student randomly interacts with the other co-students to enhance their knowledge. A student gains more knowledge if the co-student has more knowledge than him/her. Mathematically, at an iteration i, Let X_i and X_j be two different students, $i \neq j$. The new solutions are updated as follows

$$Y_{new\,i} = Y_{oldi}+r_i(Y_i-Y_j) \quad \text{if } F(Y_i) <= F(Y_j) \tag{4}$$

$$Y_{newi} = Y_{oldi}+r_i(Y_j-Y_i) \quad \text{if } F(Y_j) < F(Y_i) \tag{5}$$

Accept $Y_{new\,i}$ if it generates a better makespan value.

C. Steps of TLBO algorithm

1. Initialize the population randomly.

2. Find the best student of the population and make it the teacher. Now calculate the mean result of the total class.

3. Now calculate the difference mean using the formula of equation 1 by using the teaching factor from equation 2.

4. Update the student's value with the teacher's value using the equation 3.

5. Now update the student's knowledge with the interaction of fellow student's knowledge using the equations 4 and 5.

6. Repeat the steps 2 to 5 till the termination criterion is reached.

IV. PROBLEM FORMULATION

The formulation of FJSSP is as follows. There is a set of m machines k =1,2,…,m and a set of n jobs i=1,2,…,n. Each job i consists of a sequence of n_i operations. Processing of each operation O_{ij} (j = 1; 2…, n_i) of job i has to be done on single machine M_k out of a set of given available machines M_{ij} (for $M_k \in M_{ij}$, $M_{ij} \subseteq M$). Each operation needs single machine picked from a set of capable machines. Further, FJSSP requires to set starting and ending times of each operation. The FJSSP needs to find both allotment and an operations sequence on the machines so that given criteria is satisfied. If there is at least one operation $M_{ij} \subset M$, it is partial flexibility FJSSP (P-FJSSP); where as if $M_{ij} = M$ for every operation, it is total flexibility FJSSP (T- FJSSP).

Following assumptions are made during the solving process of FJSSP problem. Any operation should not be disrupted during its processing (non-preemptive condition). At a time, maximum only single operation can be performed on a machine (resource constraint). For any pair of operations in a job, the precedence constraints of the operations cannot be violated. Move times of jobs and setting up time of machines are negligible. Jobs are different from one another. Machines are different from one another. The aim of the FJSSP is to allot each operation to a suitable machine and then sequence all operations of each machine so that makespan can be minimized.

The notation used in this paper is as follows

n	= total number of jobs
m	= total number of machines
J_i	= total number of operations of job i
O_{ij}	= the j th operation of job i
M_{ij}	= the set of machines available for O_{ij}
P_{ijk}	= processing time of O_{ij} on kth machine
S_{ijk}	=begining time of of O_{ij} on kth machine
C_{ij}	= finish time of O_{ij}
h, i	= index of jobs, h, i = 1, 2, . . . ,n
k	= index of machines, k = 1, 2, . . . ,m
g, j	= index of operation sequence, g, j = 1, 2,. . . ,J_i
C_k	= finish time of M_k
W_k	=workload of M_k
L	=a large number
X_{ijk}	={ 1, if machine k is selected for the O_{ij} 0, otherwise}
Z_{ijhgk}	={ 1, if O_{ij} precedes O_{hg} on machine k 0, otherwise}

Manne [27] proposed the following mixed integer programming (MILP) formulation for the FJSSP.

Objective is to minimize

$$C_{max} \geq C_i \; \forall \; i \qquad (6)$$

Subject to

$$C_i \geq \Sigma C_{ijk} \; \forall \; i, j = J_i \; \& \; k \in M_{ij} \qquad (7)$$

$$S_{ijk} + C_{ijk} \leq X_{ijk} * L \forall \; i,j,k \in M_{ij} \qquad (8)$$

$$C_{ijk} \geq S_{ijk} + P_{ijk} - (1 - X_{ijk}) * L \; \forall \; i,j,k \in M_{ij} \qquad (9)$$

$$S_{ijk} \geq C_{jgk} - (Z_{ijhgk}) * L$$

$$\forall \; i \leq h, \forall \; j \; \& \; \forall \; k \in M_{ij} \cap M_{hg} \qquad (10)$$

$$S_{ijk} \geq C_{jgk} - (1 - Z_{ijhgk}) * L$$

$$\forall \; i \leq h, \forall \; j \; \& \; \forall \; k \in M_{ij} \cap M_{hg} \qquad (11)$$

$$\Sigma \; S_{ijk} \geq \Sigma \; C_{ij-1\,k} \;\; \forall \; i, \forall \; j = 2,\ldots, J_i \;\; \& \; k \in M_{ij} \qquad (12)$$

$$\Sigma \; X_{ijk} = 1 \; \forall \; i,j \; \& \; k \in M_{ij} \qquad (13)$$

$$S_{ijk} \geq 0, \; C_{ijk} \geq 0 \;\; \forall \; i,j,k \qquad (14)$$

Constraint (6) determines makespan. Completion times of jobs is determined by the constraint set (7). Constraint sets (8) & (9) conveys that difference between the completion and starting time is more than or equal to the processing time on machine k. Constraint set (10) & (11) assures that operation O_{ij} and O_{hg} are not done on the same machine at a time. Constraint (12) assures that precedence relations between the operations are properly followed. Constraint (13) assures that an operation is carried out only one machine. The constraint (14) indicates that starting and completion times of any operation on a machine k is zero it is not allotted to machine k.

V. PROBLEM MAPPING MECHANISM

We choose the following example to explain this approach with total flexibility. The task is to carry out 3 jobs on 4 machines. The data to the left hand side in table I includes operations, jobs and processing times on corresponding machines. The data to the right hand side is of priority order.

TABLE I. Example problem with priority order

Job No.	Operation No.	Machine				Priority order			
		M1	M2	M3	M4	1	2	3	4
1	O_{11}	4	5	6	2	M4	M1	M2	M3
	O_{12}	7	6	3	5	M3	M4	M2	M1
2	O_{21}	3	1	1	4	M2	M3	M1	M4
	O_{22}	5	3	2	9	M3	M2	M1	M4
	O_{23}	1	4	1	2	M1	M3	M4	M2
3	O_{31}	3	1	8	2	M2	M4	M1	M3
	O_{32}	2	3	4	1	M4	M1	M2	M3

The real number encoding system is used. Each operation is a subject of student. Each subject value of the student is denoted by a real number. The integer part of the subject value is used for allotting the machines from priority order. The allottment is shown in the table II.The fractional part of subject value is used for sequencing the operations on same machine in the ascending order of fractional part. The sequencing is shown in the table III.

TABLE II. A stochastic subject value representation.

Operation	O_{11}	O_{12}	O_{21}	O_{22}	O_{23}	O_{31}	O_{32}
Subject value	2.7732	1.9201	3.1581	2.4382	2.7889	1.2961	1.6293
Priority	2	1	3	2	2	1	1
Machine allotted	M1	M3	M1	M2	M3	M2	M4

Table III. Sequenced operations

M1	O_{21}	O_{11}
M2	O_{31}	O_{22}
M3	O_{23}	O_{12}
M4	O_{32}	

VI. LOCAL SEARCH

After the teaching learning based optimization step, the following local search method is proposed.

Step 1.At every iteration, each solution with minimum makespan is considered for local search. All the critical operations of that solution are to be found. If two or more critical operations are there adjacent to each other they need to be swapped and checked for new makesapn. If it gives a best value, replace it with original solution.

Step 2. Now each critical operation is to be checked changing the priority order of machine allotted from 1 to 3 leaving its previously allotted priority number as follows explained with examples.

For operation O11 subject value is 2.7732. Now it is to be replaced with 1.7732 and makespan is calculated next the same value replaced with 3.7732 and makespan is calculated. Similarly for operation O12 subject value is 1.9201. Now it is to be replaced with 2.9201 and makespan is calculated next the same value replaced with 3.9201 and makespan is calculated.

In this way, each critical operation is checked for priority order 1 to 3 to find the best neighbor solution.

Step 3. To avoid trapping at the local optima and to maintain the diversity, the mutation operation from genetic algorithm is incorporated to the TLBO. In the proposed algorithm, for every 10 iteration, a student is randomly picked using a rand function and it replaced with a random solution.

VII. RESULT & DISCUSSION

Experiments have been conducted on all the five Kacem's instances [3-4] from the literature using proposed TLBO. Here integrated approach is adopted i.e. by addressing routing and sequencing sub problems simultaneously. After conducting experiments to Kacem's instances from the literature, it is found that this approach using the TLBO algorithm gives satisfactory results. In the figures 1&2, x-axis indicate time units and y-axis indicates machine number.

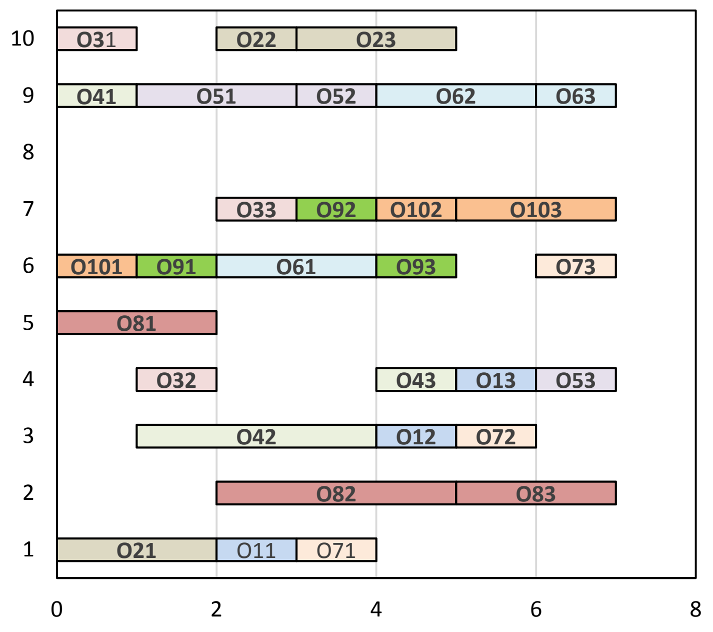

Figure1. Gantt chart of Kacem instance (10 x10 problem) obtained by TLBO

Note: Here we can observe that no operation is allotted to M8

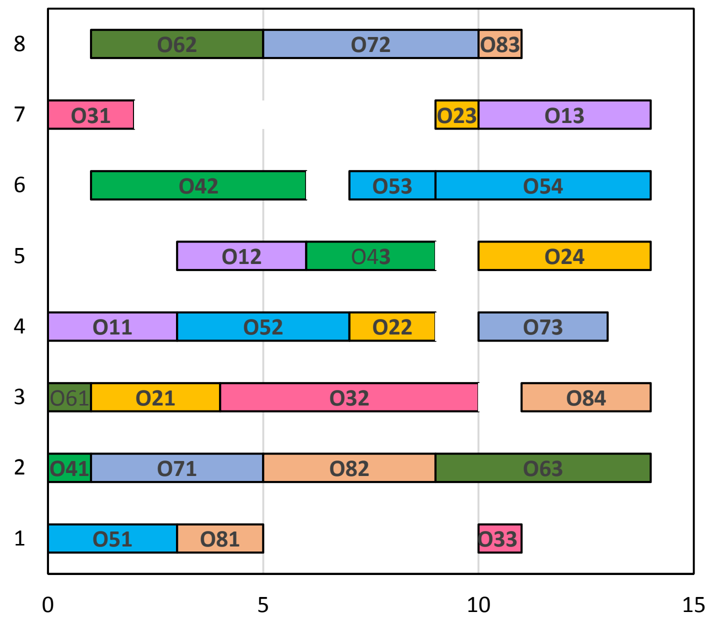

Figure2. Gantt chart of Kacem instance (8 x 8 problem) obtained by TLBO

TABLE 4. Comparison of obtained TLBO results with other algorithms

SI. No	**PROBLEM SIZE**	**AL+ CGA [4]**	**PSO +SA [12]**	**HACO [13]**	**Xing's algorithm [15]**	**AIA [7]**	**HTSA [16]**	**ABC [17]**	**BBO [18]**	**HHS [8]**	**DABC [28]**	**MOPSO [9]**	**QPSO [10]**	**Proposed TLBO**
1	4×5	16	NA	**11**	**11**	NA	**11**	**11**	**11**	NA	**11**	NA	NA	**11**
2	8×8	15	15	NA	**14**	**14**	**14**	**14**	**14**	**14**	**14**	**14**	**14**	**14**
3	10×7	NA	NA	**11**	**11**	NA	**11**	**11**	NA	NA	**11**	NA	NA	**11**
4	10×10	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**	**7**
5	15×10	23	12	12	**11**	**11**	**11**	**11**	12	**11**	**11**	**11**	**11**	12

VIII. CONCLUSIONS

Results show that the proposed TLBO has given satisfactory results. The main advantage of TLBO algorithm is that there is only one algorithmic specific parameter that needs to be tuned to get the optimal results. The additional benefit of it is that, the tuning parameter called teaching factor can be taken randomly which is not the case in other optimization techniques. Further a new efficient local search method is proposed in this work that can be applied with any optimization technique in the future. Swapping technique and mutation techniques are incorporated in the local search algorithm. The study can be extended in the future to apply modifications in the local search technique. The study can also be extended to hybridize TLBO with algorithms like TS and SA. Finally, from the solution of 10 × 10 problem, it is observed that no operation is allotted to M8 and still an optimal solution is obtained. This is for the first time in the history of FJSSP a solution is obtained in this manner for 10 × 10 problem.

References

[1] Garey MR, Johnson DS, Sethi R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research,* 1:117–129.

[2] Brandimarte P. (1993). Routing and Scheduling in a flexible job shop by tabu serch. *Annals of Operations Research*, 41, 157-183.

[3] Kacem I, Hammadi S & Borne P. (2002a). Pareto optimality approach for flexible job shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic, *Mathematics and Computers in Simulation*, 60, 245-276.

[4] Kacem I, Hammadi S & Borne P. (2002b). Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man and Cybernetics*, part C, 32(1), 408-419.

[5] Chen H, Ihlow J, Lehmann C. A genetic algorithm for flexible job-shop scheduling. InRobotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on 1999 (Vol. 2, pp. 1120-1125). IEEE.

[6] Jia HZ, Nee AY, Fuh JY, Zhang YF. A modified genetic algorithm for distributed scheduling problems. Journal of Intelligent Manufacturing. 2003 Jun 1;14(3-4):351-62.

[7] Bagheri A, Zandieh M, Mahadavi I & Yazdani M. (2010). An artificial immune algorithm for flexible job-shop scheduling problem. *Future Generation Computer Systems*, 26, 533-541.

[8] Yuan Y, Xu H & Yang J. (2013). A hybrid harmony search algorithm for the flexible job shop scheduling problem. *Applied Soft Computing*, 13, 3259-3272.

[9] Singh MR, Singh M, Mahapatra SS & Jagadev N. (2015). Particle swarm optimization algorithm embedded with maximum deviation theory for solving multi objective flexible job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 1-14

[10] Singh, M. R., & Mahapatra, S. S. (2016). A quantum behaved particle swarm optimization for flexible job shop scheduling. *Computers & Industrial Engineering*, *93*, 36-44.

[11] Mastrolilli M and Gambardella LM. (2000). Effective neighborhood functions for the flexible job shop problem. *Journal of Scheduling*, 3(1), 3-20.

[12] Xia W & Wu Z. (2005). An effective hybrid optimization approach for multi objective flexible job-shop scheduling problems. *Computers and Industrial Engineering*, 48(2), 409-425.

[13] Liouane N, Saad I, Hammadi S & Borne P. (2007). Ant systems & local search optimization for flexible job-shop scheduling production. *International Journal of Computers, Communications & Control*, 2, 174-184.

[14] Xing L, Chen Y & Yang K. (2009a). Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. *Applied Soft Computing*, 9, 362-376.

[15] Xing L, Chen Y & Yang K. (2009b). An efficient search method for multi objective flexible job shop scheduling problems. *Journal of Intelligent Manufacturing, 2*, 20,283-293.

[16] Li J, Pan Q & Liang Y. (2010). An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 59, 647-662.

[17] Wang L, Zhou G, Xu Y, Wang S & Liu M. (2012). An effective artificial bee colony algorithm for the flexible job-shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, 60,303-315.

[18] Habib S, Rahmati A & Zandieh M. (2012). A new biogeography-based optimization (BBO) algorithm for flexible job shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, 58, 1115-1129.

[19] Gao KZ, Suganthan PN, Pan QK, Chua TJ, Cai TX, & Chong CS. (2014). Discrete harmony search algorithm for flexible job shop scheduling problem with multiple objectives. *Journal of Intelligent Manufacturing*, 1-12.

[20] Sun L, Lin L, Wang Y, Gen M & Kawakami H. (2015). A Bayesian Optimization-based Evolutionary Algorithm for Flexible Job Shop Scheduling. *Procedia Computer Science*, 61, 521-526.

[21] Chang HC, Chen YP, Liu TK & Chou JH. (2015). Solving the Flexible Job shop scheduling problem with makespan Optimization by using a Hybrid Taguchi- Genetic Algorithm. *Access, IEEE*, 3, 1740-1754.

[22] Zhang G, Gao L, and Shi Y. (2011). An effective genetic algorithm for the flexible job-shop scheduling problem *Expert Systems with Applications*, 38(4), 3563-3573.

[23] Singh MR & Mahapatra SS. (2012). A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks. *International Journal of Advanced Manufacturing Technology*, 62, 267-277.

[24] Rao RV, Savsani VJ and Vakharia DP. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303-315.

[25] Rao RV & Patel V. (2012). An elitist teaching – learning – based optimization algorithm for solving complex constrained optimization problems. *International Journal of Industrial Engineering Computations*, 3(4), 535-560.

[26] Rao RV & Patel V. (2013). Comparative performance of an elitist teaching – learning – based optimization algorithm for solving unconstrained optimization problems. *International Journal of Industrial Engineering Computations*, 4(1), 29-50.

[27] Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, *8*(2), 219-223.

[28] Li, J. Q., Pan, Q. K., & Tasgetiren, M. F. (2014). A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities. *Applied Mathematical Modelling*, *38*(3), 1111-1132.