

Ant Colonized and Taguchi Parallel Scheduling with Sequence Independent Setup Time



G.Geetha, T.Hemamalini

Abstract The Job Shop Parallel Machine Scheduling (JSPMS) is a hybrid production system, and hence has received significant attention in the past few years. The JSPMS problem is a rationalization of the traditional job shop scheduling problem in computer science and operation research that permits to process operations on single machine out of a set of possible parallel machines. To maximize the job completion rate and minimize job completion time, a hybrid production system is necessary. With this objective, a novel meta-heuristic method is designed. This paper develops a scheduling method called, Ant Colonized and Taguchi Parallel Operation Scheduling (AC-TPOS), for JSPMS, aimed to minimize job completion time and maximize job completion rate. The design of AC-TPOS method involves two different models, namely, Ant Colonized Parallel Machine Selection (ACPMS) model and Taguchi Parallel Operation Scheduling (TPOS) model. In ACPMS model, optimal selection of machine is done via operation being processed by parallel machines using local pheromone updating rule concentrating on the makespan time. In addition, the processing time and sequence-independent setup time are considered. Next, in TPOS model, optimal scheduling of operation is performed using Taguchi method concentrating on the makespan rate. Finally, the test results first show that our algorithm outperforms existing methods in terms of job completion rate, job completion time and computational complexity involved in scheduling operations.

Keywords: Job Shop, Parallel Machine, Scheduling, Ant Colonized, Taguchi, Processing time, Setup time

I. INTRODUCTION

Scheduling problems exist in several economic domains. Some of them are, airplane scheduling, tour scheduling, train scheduling, and specifically in the manufacturing shop scheduling. Scheduling is considered to be the most crucial aspects to enhance the organization productivity and satisfy the deadlines. Therefore efficient scheduling has become compulsory to every organization to sustain the goodwill between their consumers and continuity. The Flexible Job Shop (FJS) refers to a hybrid production system, and has drawn substantial attention in the past few years. A Teaching Learning Based Optimization (TLBO) was presented in [1] to address the issues related to Flexible Job Shop Scheduling Problem (FJSP) based on the integrated approach. But, like several meta-heuristics model, TLBO also has a propensity to get confined at local optimum.

To address this situation, a local search technique in addition to mutation strategy from genetic model was included to TLBO to enhance the solution quality and diversity maintenance in the population. Though the enhancement of solution quality with diversity maintenance was assured, less focus was done on job completion time. To address this issue, Ant Colonized Parallel Machine Selection (ACPMS) model is designed in this work that introduces local pheromone updating rule ensuring minimum job completion time.

Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Times (FJSP-SDST) was presented in [2] to reduce makespan. A mathematical model to solve small instances for FJSP-SDST to optimality was presented. Next, with the aid of disjunctive graph model, a tabu search algorithm was designed with specific neighborhood functions and a diversification structure, therefore resulting in minimum computational time for same class of problem instances. Though computational time with the aid of tabu search algorithm was said to be reduced, but using sequence dependent setup time, both job to be processed and prior job processed have to be considered, that minimized job completion rate. To improve the job completion rate, a sequence-independent setup type is considered along with the taguchi method.

In this paper, we present how to solve the Job Shop Parallel Machine Scheduling (JSPMS) problem by hybridization of two meta-heuristic models. The proposed method is developed into two major models. The machine selection model (MSM) assigns operations to machines using Parallel Colonized Machine Selection algorithm, while the Operation Scheduling Module (OSM) determines the processing order of operations on machines using the Dynamic Taguchi Operation Scheduling algorithm. The objectives of proposed algorithms are the maximum job completion rate, minimum job completion time and computational complexity.

This paper is organized as follows. The next Section 2 reviews the relevant literature. Section 3 introduces with the preliminaries introduced. Section 4 describes the detailed procedure of the proposed method. Next, Section 5 shows the results and presents a discussion. Finally, Section 6 draws conclusions.

II. RELATED WORKS

Scheduling is a scientific domain related to the assignment of limited operations over time. The objective of scheduling is to maximize job completion rate, time and so on. A general modeling method was introduced in [3] for job shop scheduling problems. Job-shop scheduling problem contains both batches and anticipatory sequence-dependent setup times by timed Petri nets. But, the job completion rate was sufficient.

Revised Manuscript Received on February 05, 2020.

* Correspondence Author

G.Geetha, Research Scholar, Department of Mathematics, Government Arts College, Coimbatore-18.

Dr T.Hemamalini, Assistant professor, Department of Mathematics, Government Arts College, Coimbatore-18.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

A novel method that integrated both particle swarm optimization (PSO) and iterated local search (ILS) was presented in [4] to address issues related to hybrid flowshop scheduling (HFS) problem. However, multi resources were not taken into consideration.

In [5], Differential Evolution and Greedy Algorithm (DEGA) were applied with the objective of improving robustness for multi-skill resource-constrained problem. Yet another method to solve robust job shop scheduling was presented in [6] by applying problem-specific neighborhood structure. Manufacturing of semiconductor is considered to be the most time consuming job shop scheduling problem. Two different operators namely, Cauchy distribution and genetic operators were introduced in [7], therefore improving the time for scheduling. A Discrete Jaya algorithm was presented in [8] for flexible job shop ensuring rescheduling.

Timely solving of job shop problem is considered to be very cumbersome and heavily required in the domains of management, industry and economy. In [9], a discrete Imperialistic Competition Algorithm (ICA) and Tabu Search (TS) were presented using Fuzzy Job Shop Scheduling (FJSS) to minimize the completion time involved. In literature for mathematical model, scheduling problems were addressed by either following proper routing or process plan flexibility, but not both. In [10], mathematical modeling involving both routing and process plan flexibility using mixed-integer programming was presented. The computational efficiency was found to be improved. A review of flexible job scheduling techniques was presented in [11]. Scheduling of workshop has primarily concentrated on performances involving times and quality. However, in recent years, environmental metrics have received a significant impact. In [12], an energy-efficient job shop scheduling method, called, Grey Work Optimization Double-searching Mode (DMGWO) was presented to reduce the total cost of energy-consumption and tardiness. However, idle time involved was not considered. To address this issue in [13], two-level metaheuristic algorithm was designed, comprising of both lower level algorithm and upper level algorithm. However, with occurrence of uncertain events in real world situations, normal scheduled activities were said to be interrupted. Considering this aspect, in [14], a two-stage Teaching Learning Based Optimization (TLBO) method was designed that also considered the machine break down data to conventional fast job scheduling, therefore ensuring robustness and stable predictive schedules. One of the complex combinatorial optimization methods is the permutation flow shop problem. Though greater amount of algorithms have been proposed to address the static permutation flow shop problems, however, permutation flow shop problems are not static but are dynamic. In [15], a memetic algorithm based scheduling approach was designed to address multiple order permutation. A case study using Nash bargaining model for tire manufacturing in Iran was presented in [16]. However, in the above two methods, the processing time were not found to be controllable. To address this issue, a policy was designed to evaluate the workload of the resources. Here, two approaches were presented in [17], where reduction of processing time was considered first and then additional resources were distributed evenly. With this the processing time was found to be controlled. However, the setup time required was not considered. In [18], a sequence dependent

setup time for job shop scheduling using heuristics method was presented. Yet another job scheduling approach with sequence dependent setup time and preventive maintenance was presented in [19]. An optimization model using taguchi was presented in [20] to obtain the controllable parameters involved in scheduling. In [20], a Distributed Ant Colony System (DACS) approach was introduced for multi-objective optimization in production scheduling under a flexible job shop environment with assembling operations. However, the performance of computational complexity was not considered.

A kind of scheduling problem with fuzzy processing times and fuzzy resource constraint on uniform parallel machines was designed in [22] to minimize the makespan given that the total fuzzy resource utilization cannot exceed a given fuzzy number. However, the performance enhancement of job completion rate was not sufficient. With an inspiration from the above works, an Ant Colonized and Taguchi Parallel Operation Scheduling (AC-TPOS) method to solve JSPMS, under processing time and setup time and study its scheduling performance is presented.

1. Preliminaries

Let us consider a scheduling problem [3] defined by three factors $\{\alpha, \beta, \gamma\}$. Here, among the three factors, the ' α ' field corresponds to the machine domains. Next, the ' β ' field corresponds to the organizing characteristics and limitations that may be pertinent. Finally, the ' γ ' field narrates the objective to be attained. (Minimize job completion time and maximize job completion rate), with the given constraints involving total processing time and setup time. Let us further assume that all the jobs arrive at the shop at time '0'. The problem considered in this paper is defined as follow.

$$\alpha \rightarrow J_m \quad (1)$$

From the above equation (1), ' α ' field, corresponds to the job shop with ' m ' machines.

$$\beta \rightarrow BSize_{ij}, ST_{ij}, PT_{ij} \quad (2)$$

From the above equation (2), ' β ' field corresponds to the batch size ' $BSize_i$ ', setup time ' ST_{ij} ' and processing time ' PT_{ij} ' respectively. Here, the batch size of the jobs are said to be processed at one time belonging to the same ' ith ' family, therefore ensuring parallel machine flow. The setup time for a machine ' k ' precedes processing family ' i '. A definite setup time is only required for machine ' k ' if the former processed batch does not apply to family ' i '. From the above equation (2), ' ST_{ij} ' represents the setup time required for processing family ' j ' on machine ' k ' directly after family ' i ' has been processed on it.

Finally, the processing time ' PT_{ij} ' belongs to the ' ith ' family on ' jth ' machine respectively. Let us consider that the sequence independent type here denotes the setup depending on the job to be processed ' J_i '. On the other hand, sequence dependent denotes the setup depending on both the job to be processed ' J_i ' and the prior job ' J_{i-1} '. In this paper, the sequence-independent setup type is considered, and the setup time ' ST_{ij} ' is included in the processing time. Here, total processing time refers to the summation of setup time ' ST_{ij} ' and processing time ' PT_{ij} '.

$$\gamma \rightarrow \{Min(JCT)Max(JCR)\} \quad (3)$$

From the above equation (3), ‘ γ ’ field corresponds to the actual objective to be achieved. In this work, two objectives must be achieved, i.e. minimizing the job completion time and maximizing the job completion rate.

2. Ant Colonized and Taguchi Parallel Operation Scheduling (AC-TPOS) method

Searching for solutions to solve the Job Shop Parallel Machine Scheduling (JSPMS), a method called, Ant Colonized and Taguchi Parallel Machine Operation Scheduling (AC-TPOS) is presented. The AC-TPOS method proposes an algorithm in which a Taguchi Improved Ant Colony Optimization method is embedded in the evolution phase to facilitate high-quality pheromone intensity and effectively increase the convergence speed. Therefore, the proposed method is more likely to identify a near optimal solution and avoid local optimal solutions. This is performed using two different models. They are Ant Colonized Parallel Machine Selection model and Taguchi Parallel Operation Scheduling model. The Ant Colonized Parallel Machine Selection model helps an operation to select one of the parallel machines to process it. The Taguchi Parallel Operation Scheduling model then schedules the sequences and date of all operations. Figure 1 given below shows the block diagram of Ant Colonized and Taguchi Parallel Operation Scheduling (AC-TPOS) method.

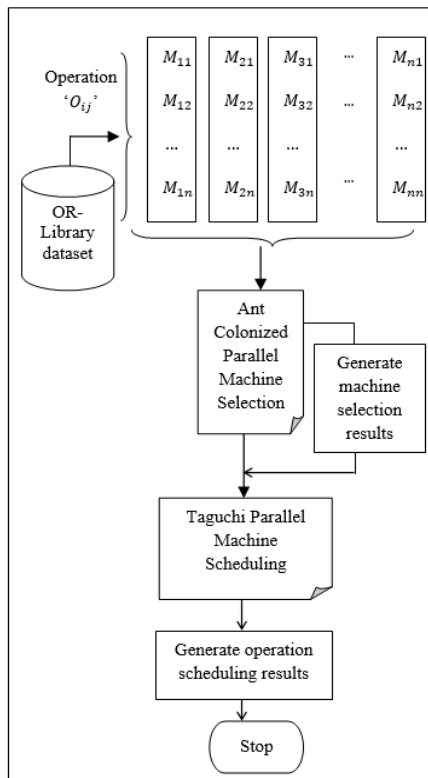


Figure 1 Block diagram of Ant Colonized and Taguchi Parallel Operation Scheduling

As illustrated in the above block diagram, the Ant Colonized and Taguchi Parallel Operation Scheduling (AC-TPOS) method involves two different steps. They are machine selection in a parallel manner using the improved ant colony model and machine scheduling using the taguchi model. While applying the Parallel Colonized Machine Selection algorithm, quantity of pheromone, selection of next node and process of updating the pheromone are the three things analyzed. With the appropriate number of pheromone

selected using makespan and selection of next node done based on neighbors, the job completion time is said to be reduced. Next, with the optimal machines being selected, the operations involving several jobs has to be scheduled in such a manner that maximum job completion rate is ensured. In this work, a Taguchi Operation Scheduling model is applied to the optimal machines being selected. With this, inappropriate scheduling of operations between machines is reduced, therefore maximizing the job completion rate. The elaborate description is discussed in forthcoming sections.

2.1 Ant Colonized Parallel Machine Selection model

In this section, an Ant Colonized Parallel Machine Selection model is designed with the objective of helping an operation to identify one of the parallel machines to process it with minimum makespan (i.e. job completion time), taking into account the processing time and setup time. Here for each operation the selection of machines is done in a parallel manner. The Job Shop Parallel Machine Scheduling (JSPMS) is formulated as follows. Let us consider a set of ‘ n ’ jobs ‘ $J = \{J_1, J_2, \dots, J_n\}$ ’ to be processed on a set of ‘ m ’ machines ‘ $M = \{M_1, M_2, \dots, M_m\}$ ’. Each job ‘ J_i ’ is formed by a sequence of ‘ ij ’ operations ‘ $O_{11}, O_{12}, \dots, O_{ij}$ ’ to be performed successively according to the sequence ‘ S ’. For each operation ‘ O_{ij} ’, there is a set of alternative machines ‘ $M(O_{ij})$ ’ capable of performing it. Figure 2 shows the flow diagram of Ant Colonized Parallel Machine Selection model followed in our work.

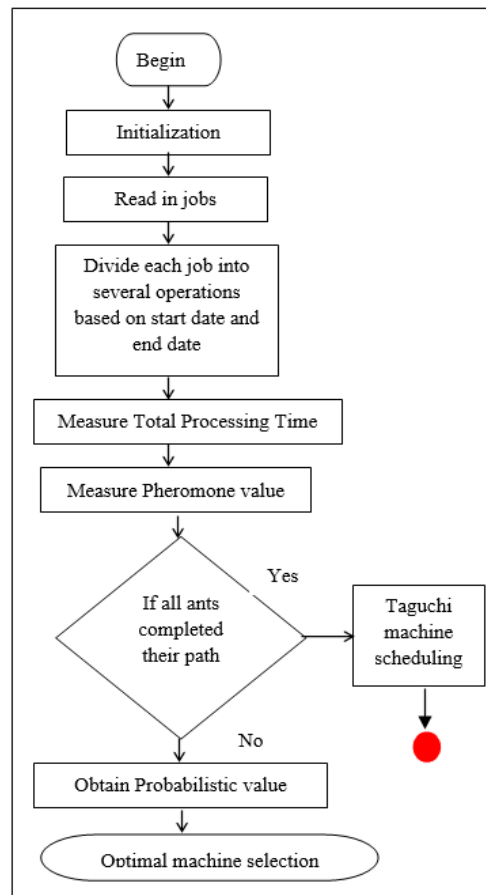


Figure 2 Flow diagram of Ant Colonized Parallel Machine Selection model

Ant Colonized and Taguchi Parallel Scheduling with Sequence Independent Setup Time

From the above figure, for each operation ‘ O_{ij} ’, with four machines ‘ M_k ’ (i.e. with ‘ $k = 4$ ’) said to exist ‘ M_1, M_2, M_3, M_4 ’, in the machine selection model, machines are selected in a parallel fashion by applying Ant Colonized Parallel Machine Selection model [21]. Then, the Flexible Job Shop problems considered in this work with parallel machine is shown using the classical notation as given below.

$$N \rightarrow (J_i, SDate_{ij}, EDate_{ij}, TPT_{ij}, ST_{ij}, PT_{ij}, S) \quad (4)$$

From the above equation (4), the classical notation used in the proposed works includes, ‘ J ’ job, with start date and end date represented by ‘ $SDate_j$ ’ and ‘ $EDate_j$ ’, processing time and total processing time denoted by ‘ PT_{ij} ’

and ‘ TPT_{ij} ’, for each corresponding sequence ‘ S ’ with the setup time taken into account as ‘ ST_{ij} ’ respectively. The processing of job ‘ J ’ on machine ‘ i ’ corresponds to as operation ‘ O_{ij} ’ with processing time given as ‘ PT_{ij} ’ respectively. Besides, Job ‘ J ’ has an end date ‘ $EDate_j$ ’ and a start date ‘ $SDate_j$ ’ respectively. No job can start before its start date and its processing should not exceed its end date. If operation ‘ O_{ijk} ’ immediately succeeds operation ‘ O_{ij} ’ on machine ‘ i ’, we assume in the proposed work that a setup time ‘ ST_{jk}^i ’ is said to occur. Table 1 given below shows instance of classical notation problem with four jobs and four machines.

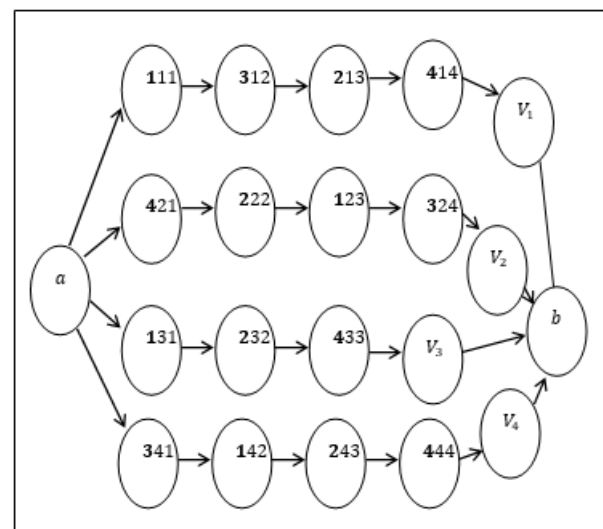
Table 1 Sample notation

Job	Start Date $SDate_{ij}$	End Date $EDate_{ij}$	Total Processing Time $[TPT_{ij}]$	Sequence	Setup Time ST_{ij}	Processing Time PT_{ij}
1	0	41	37	1 – 3 – 2 – 4	2	$PT_{11} = 9, PT_{31} = 9, PT_{21} = 10, PT_{41} = 7$
2	0	33	21	4 – 2 – 1 – 3	3	$PT_{42} = 8, PT_{22} = 4, PT_{12} = 3, PT_{32} = 3$
3	3	35	30	1 – 2 – 4	2	$PT_{13} = 17, PT_{23} = 6, PT_{43} = 5$
4	1	39	22	3 – 1 – 2 – 4	1	$PT_{34} = 7, PT_{14} = 4, PT_{24} = 5, PT_{44} = 5$

As given in the above table 1, let us consider four jobs ‘ J_1, J_2, J_3, J_4 ’ with start date of ‘0, 0, 3, 1’ and end date being ‘41, 33, 35, 39’ respectively. Let us further consider the sequence ‘ S ’ being ‘1 – 3 – 2 – 4’, ‘4 – 2 – 1 – 3’, ‘1 – 2 – 4’ and ‘3 – 1 – 2 – 4’ for four different jobs being processed parallel in four different machines. In this work, further we have assumed that the total processing time ‘ TPT_{ij} ’ is equal to the processing time ‘ PT_{ij} ’ and the setup time ‘ ST_{ij} ’ and mathematically expressed as given below.

$$TPT_{ij} = PT_{ij} + ST_{ij} \quad (5)$$

In ACO, a set of software agents called ants (i.e. operations) search for good solutions (i.e. selection of optimal machine in a parallel machine) is performed. These initial solutions are created randomly. For each machine ‘ M_{ij} ’ belonging to different machine group ‘ M_k ’, a sequence ‘ S ’ is thus randomly generated. This is because for parallel machines, the machine selection and scheduling are randomly performed by balancing the work-load of the machine. With the above said considerations, by applying improved ant colony optimization, a solution is generated, as illustrated in the figure given below (i.e., from ‘ a ’ to ‘ b ’). The passage is constructed by an ant node by node.



Sample to be included

Figure 3 Ant Colonized Parallel Machine Selection

As illustrated in the above figure, four different passages are constructed ‘1 – 3 – 2 – 4’, ‘4 – 2 – 1 – 3’, ‘1 – 2 – 4’ and ‘3 – 1 – 2 – 4’ respectively. The objective here is to motivate the ants walking through the graph (i.e. different machine group), at each node (i.e. single machine group) they have to choose one arc (i.e. single machine at a time).

For example in the above figure, with single machine group ‘111’, ‘312’, ‘213’, ‘414’, where ‘1-3-2-4’ represents the corresponding sequence for the first machine ‘1’, followed by which four different machines being checked one by one ‘1, 2, 3, 4’. In a similar manner, with the second single machine group ‘421’, ‘222’, ‘123’, ‘324’, where ‘4-2-1-3’ represents the corresponding sequence for the second machine ‘2’ followed by which four different machines being checked one by one ‘1, 2, 3, 4’. This probability increases with the increase in the number of ants (parallel machines in existence) that are further assumed by the pheromone lay down by each ant. In this manner, the entire solution is created based on ant colony, for the above sample four different sequences and different machines being processed in a parallel fashion.

Besides, three things are analyzed in the above Improved Ant Colony Optimization, with the objective of minimizing the makespan (i.e. minimizing job completion time). They are the quantity of pheromone on the arc ‘ $P_{a,b}$ ’, selection of next node ‘ b ’, and finally how the pheromone quantity is updated. In this work, the quantity of pheromone on the arc is updated based on the makespan. This is formulated as given below.

$$P_{a,b} = \sum_{k=1}^n P_{a,b}^k \quad (6)$$

$$P_{a,b}^k = \begin{cases} \frac{C}{MS_k}, & \text{if arcs } (a \rightarrow b) \text{ belongs to the path of ant } k \\ 0, & \text{Otherwise} \end{cases} \quad (7)$$

From the above equation (6) and (7), the quantity of pheromone on the arc ‘ $P_{a,b}$ ’ is determined by the makespan ‘ MS ’ selected by the ant ‘ k ’, with ‘ C ’ referring to the constant and ‘ n ’, referring to the number of jobs generated. Second, the selection of next node is done based on probabilistic solution. The probability of a solution from ‘ $Neigh(s^p)$ ’ is performed in a probabilistic manner at each construction step. The exact rules for the probabilistic choice of solution vary across different ACO variants and the best known rule is formulated as given below.

$$Prob(O_{ij} | s^p) = \frac{[P_{ab}][MST_{ab}]}{\sum_{O_{ij} \in Neigh(a)} [P_{ab}][MST_{ab}]}, O_{ij} \in Neigh(a) \quad (8)$$

From the above equation (8), ‘ P_{ab} ’ refers to the pheromone value and ‘ MST_{ab} ’ refers to the heuristic value, where, ‘ MST_{ab} ’ correspond to an estimate of desirability of the transition ‘ a, b ’ according to the apparent makespan time, ‘ $Neigh(a)$ ’ represents the neighboring nodes (machines) to ‘ a ’, for quick convergence of all ants. After making a choice for an arc, the pheromone values are updated based on the Taguchi method (discussed in the next section). The pseudo code representation of Parallel Colonized Machine Selection algorithm is given below.

Algorithm 1 Parallel Colonized Machine Selection algorithm

Input: Operation ‘ O_{ij} ’, Job ‘ J_i ’, Start Date ‘ $SDate_{ij}$ ’, End Date ‘ $EDate_{ij}$ ’, Sequence ‘ S ’, Processing Time ‘ PT_{ij} ’, Setup Time ‘ ST_{ij} ’, Total Processing Time ‘ TPT_{ij} ’

Output: Optimal machine selection with minimal makespan cost (minimum job completion time)

1: **Begin**

2: **For** each Operation ‘ O_{ij} ’

3: Measure Total Processing Time using equation (5)

4: Measure the Pheromone value using equation (6)

5: Obtain probabilistic choice of solution using equation (8)

6: **Return** (optimal machine selected)

7: **End for**

8: **End**

As given in the above algorithm, with a given set of jobs and each job comprising of a set of operations that have to be processed in a parallel manner, each operation has to be processed on a definite machine with a deterministic processing time, considering the setup time. In first step, total processing time of the machine is calculated. In second step, based on makespan the pheromone value is updated. Next step, the optimal machine selection is performed depends on probabilistic solution. Besides, with each machine processing one operation at a time, the objective of the above algorithm remains in selecting the optimal machine by applying improved pheromone value for quick convergence of all ants, concentrating on the makespan time, therefore minimizing the job selection time. Once, the machine selection is performed, scheduling of operations has to be done. This is performed in the work by applying the Taguchi method, explained in the next section.

2.2 Taguchi Parallel Operation Scheduling (TPOS) model

Once, the machines are selected, the corresponding operations are scheduled according to the sequence ‘ S ’ and time ‘ $SDate_{ij}, EDate_{ij}$ ’ respectively. This is done in the proposed work by applying the Taguchi Parallel Operation Scheduling (TPOS) model. The steps to be involved in TPOS are. First a two level array ‘row r and column c ’ for matrix specimens and two ants for each matrix demonstration is chosen. The size of the two level arrays is determined based on the size of the JSPMS. Table 2 given below shows the two level array representation.

Table 2 Specimen of two level array representations

Number of runs	M1				M2			M3			
	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											

Followed by the representation of matrices, S/N ratios and fitness value of all combinations in the matrix demonstrations are obtained. Next, an optimal ant based on the S/N ratio is obtained. In the proposed TPOS model, a measure of robustness is utilized to recognize how the pheromone quantity is updated that improves job completion rate by minimizing the effects of uncontrollable factors (noise factors).

Control factors are those parameters that can be controlled. However, noise factors are not said to be controllable, but can be controlled during experimentation using Taguchi method [22]. By applying Taguchi method, noise factors are manipulated with the objective of forcing variability to occur and from the results, optimal control factor to recognize the pheromone quality updates are made. Higher values of the signal-to-noise ratio (S/N) identify control factor settings that minimize the effects of the noise factors (i.e. job completion time) and therefore helps in increasing the job completion rate. The signal-to-noise ratio is formulated as given below.

$$\frac{S}{N} = -10 \log \left(\frac{\sum \frac{1}{Y^2}}{n} \right) \quad (9)$$

From the above equation (9), the signal-to-noise-ratio ' $\frac{S}{N}$ ' is measured using 'Y' corresponding to the machine waiting to be scheduled in a parallel manner and 'n' number of different machine types to be scheduled parallel.

$$C_{min} = \text{MAX} \left\{ \text{MIN} \left\{ \sum_{i=1}^n \sum_{j=1}^n (SDate_{ijk} + TPT_{ijk}) \right\} \right\} \quad (10)$$

From the above equation (10), ' C_{min} ' represents the maximal job completion rate, ' $SDate_{ijk}$ ' represents the start date of operation 'j' for job 'i' in machine 'k' and ' TPT_{ijk} ' represent the total processing time of operation 'j' for job 'i' in machine 'k'. Finally, validation is performed in such a manner that the creation of all convergence paths in the weighted graph is completed. In this manner, all convergence paths in the weighted graph are validated. The pseudo code representation of Dynamic Taguchi Operation Scheduling algorithm is given below.

Algorithm 2 Dynamic Taguchi Operation Scheduling algorithm

Input: Operation ' O_{ij} ', Job ' J_i ', Start Date ' $SDate_{ij}$ ', End Date ' $EDate_{ij}$ ', Sequence ' S ', batch size ' $BSize_i$ '
Output: Optimal Operation Scheduling with maximum job completion rate
1: Begin 2: For each Operation ' O_{ij} ' with Job ' J_i ' to be scheduled parallel 3: Repeat 4: Obtain two level array ' $row\ r\ and\ column\ c$ ' for matrix specimens 5: Measure signal-to-noise ratio using equation (9) 6: Measure fitness function using equation (10) 7: Until (all convergence paths in the weighted graph is completed) 8: Else go to step 4 9: End for 10: End

As given in the above Dynamic Taguchi Operation Scheduling algorithm, the algorithm is said to be dynamic because of three reasons. In First step, a matrix representation is applied that possess the advantage of analyzing linear operations. Second step, using the signal-to-noise ratio the effects of noise or inappropriate scheduling of operations are said to be minimized. Finally, with the measure of fitness function, the maximum job completion

rate is analyzed. Hence, the algorithm is said to be dynamic and also helps in maximizing the job completion rate.

III. NUMERICAL EXPERIMENTS

Three experiments based on the benchmark OR-Library, were performed to validate the proposed Ant Colonized and Taguchi Parallel Operation Scheduling (AC-TPOS) method and comparison was made with the two existing Teaching Learning Based Optimization (TLBO) [1] and Flexible Job Shop Scheduling Problem with Sequence Dependent Setup Times (FJSP-SDST) [2]. Three set of experiments were conducted using MATLAB. The first experiment was conducted to illustrate the convergence graph of job completion time using the TLBO [1] and FJSP-SDST [2]. The second experiment was conducted to illustrate the convergence graph of job completion rate using the TLBO [1] and FJSP-SDST [2]. Finally, the third set of experiments was conducted to evaluate the computational complexity involved in scheduling.

2.3 Job completion time

The first parameter to be measured for Job Shop Parallel Machine Scheduling (JSPMS) problem is the job completion time. The job completion time refers to the time consumed to accomplish a job. It is measured as given below.

$$JCT = \sum_{i=1}^n J_i * TPT \quad (11)$$

From the above equation (11), job completion time ' JCT ' is measured based on the number of jobs ' J_i ' and total processing time ' TPT '. Here, total processing time refers to the summation of processing time ' PT ' and the setup time ' ST '. The job completion time is measured in terms of milliseconds (ms). The graphical representation of job completion time using the three methods, AC-TPOS TLBO [1] and FJSP-SDST [2] are given below.

Table 1. Tabulation results of Job completion time

Number of jobs	Job completion time (ms)		
	AC-TPOS	TLBO	FJSP-SDST
15	0.825	1.15	1.395
30	1.135	1.455	1.855
45	1.245	1.545	1.932
60	1.355	1.585	2.135
75	1.485	1.635	2.155
90	1.5	1.855	2.325
105	1.735	2.355	2.815
120	2.125	2.515	3.035
135	2.145	2.62	3.335
150	2.235	2.686	3.855

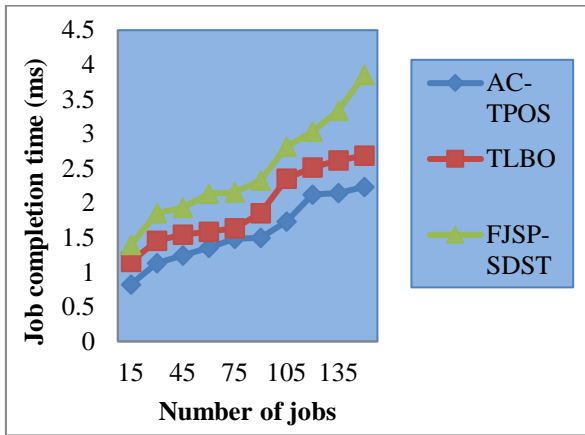


Figure 4 Convergence graph of job completion time

Figure 4 given above illustrates the convergence graph of job completion time with respect to 150 different numbers of jobs. As illustrated in the figure, the number of jobs is found to be linearly or directly proportional to the job completion time. This means that with the increase in the number of jobs, the operations to be assigned to different machines though parallel increase with the increase in the number of jobs. Due to this, the job completion time is also found to be increasing. Comparison analysis shows better results when applied with the proposed AC-TPOS method. This is because of the application of Parallel Colonized Machine Selection algorithm. By applying the Parallel Colonized Machine Selection algorithm, two main things are said to happen. First, operations are allotted or the machines are selected in a parallel fashion. Next, by applying the improved ant colony algorithm improved pheromone values are applied for quick convergence of all ants of quick convergence of all machines for operations to be performed. With this reason, the job completion time using the AC-TPOS method is found to be reduced by 19% compared to TLBO [1]. In addition, with application of deterministic processing time, considering the setup time, start of operation and completion is found to be optimal. Due to this, job completion time using AC-TPOS method is reduced by 36% compared to FJSP-SDST [2].

2.4 Job Completion Rate

The second parameter to be measured for Job Shop Parallel Machine Scheduling (JSPMS) problem is the job completion rate. Job completion rate refers to the completion of job (i.e. operations to be performed) at a stipulated time from the available machines in a parallel manner. It is formulated as given below.

$$JCR = \sum_{i=1}^n J_c / J_i * 100 \quad (12)$$

From the above equation (12), the job completion rate ‘JCR’ refers to the percentage ratio of jobs completed ‘J_c’ to the overall jobs ‘J_i’ considered for experimentation. Higher job completion rate ensures efficiency of the method. It is measured in terms of percentage (%). The graphical representation of job completion rate using three methods, AC-TPOS TLBO [1] and FJSP-SDST [2] are given below.

Table 2. Tabulation results of Job completion rate

Number of jobs	Job completion rate (%)		
	AC-TPOS	TLBO	FJSP-SDST
15	86.66	73.33	66.66
30	84.35	70.35	65.15
45	82.15	72.25	60.25
60	80.55	75.55	61.35
75	81.35	74.35	62.35
90	85.55	73.15	65.5
105	86.26	71.25	60.25
120	83.35	73.35	60.85
135	80.15	72.15	62.35
150	82.45	70.35	64.55

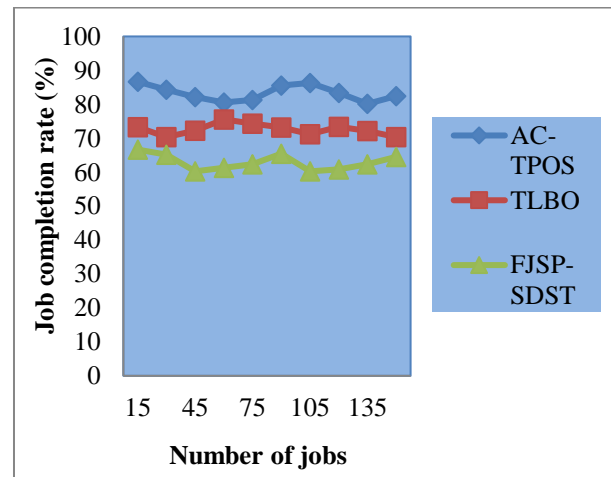


Figure 5 Convergence graph of job completion rate

Figure 5 given above illustrates the convergence graph of job completion rate. In the figure, x axis represents the number of jobs and y axis represents the job completion rate. From the figure it is illustrative that the number of jobs is non-linear in nature or inversely proportional to each other. This means that with the increase in the number of jobs neither steady increase nor decrease is found in the job completion rate. This is because of the reason that, as parallel machine scheduling is applied in our work, with the entrant of any number of jobs, job completion is said to be ensured, however, with the change in the time or rate. Experiments conducted on the above 150 different numbers of job, the job completion rate using the proposed AC-TPOS method is found to be better than TLBO [1] and FJSP-SDST [2] respectively. This is due to the reason that the machine selection is performed randomly by balancing the work-load of the machine using improved ant colony optimization. The improved ant colony optimization here refers to the three factors consideration, quantity of pheromone, selection of next node, and finally how the pheromone quantity is updated. Due to this three factor consideration in AC-TPOS method, the job completion rate is said to be improved by 15% compared to TLBO [1].

Besides, how the pheromone quantity is updated is performed by applying Taguchi method. Due to this, the job completion rate is said to be improved using AC-TPOS method by 32% compared to FJSP-SDST [2] respectively.

2.5 Computational complexity

Computational complexity involved in Job Shop Parallel Machine Scheduling (JSPMS) problem is measured. The computational complexity refers to the memory consumed to assign an operation to a specific machine. It is measured as,

$$CC = \sum_{i=1}^n J_i * MEM [O_i] \quad (13)$$

From above equation (14), computational complexity 'CC' is measured based on number of jobs 'J_i' and memory consumed in assigning an operation 'MEM [O_i']'. The computational complexity is measured in terms of megabytes (MB). The graphical representation of computational complexity using three methods, AC-TPOS TLBO [1] and FJSP-SDST [2] are given below.

Table 2. Tabulation results of Computational Complexity

Number of jobs	Computational Complexity (MB)		
	AC-TPOS	TLBO	FJSP-SDST
15	120	140	225
30	159	170	240
45	190	220	260
60	220	270	310
75	240	290	330
90	290	330	350
105	330	390	430
120	370	410	480
135	410	480	520
150	440	520	550

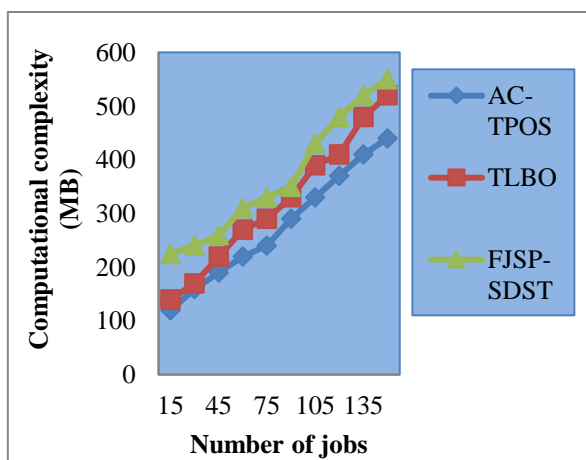


Figure 6 Convergence graph of computational complexity

Finally, figure 6 given above illustrates the graphical representation of computational complexity with respect to different jobs with varying number of machines.

Computational complexity is found to be increasing in the above figure with the increase in the number of jobs. This is due to the reason that with the increase in the number of jobs to be scheduled in different machines, the processing time, the setup time varies and hence, the computational complexity involved is also found to be increased. However, simulations experiments conducted reveals that the computational complexity involved using the proposed AC-TPOS is found to be lesser than TLBO [1] and FJSP-SDST [2]. This is due to the application of Dynamic Taguchi Machine Scheduling algorithm. By applying the Dynamic Taguchi Machine Scheduling algorithm, first, by applying the matrix representation, linear operations are addressed, that is the increase in the number of jobs to be scheduled with the specified machines. Next, by measuring the signal to noise ratio in the AC-TPOS, inappropriate scheduling of machines is said to be reduced and finally, with the application of fitness function, maximum job completion rate is ensured. Due to the above said reasons, the computational complexity using AC-TPOS is found to be reduced by 14% when compared to TLBO [1] and 27% when compared to FJSP-SDST [2] respectively.

IV. CONCLUSION

In this paper, we present a new meta-heuristic method called, Ant Colonized and Taguchi Parallel Machine Scheduling (AC-TPMS), for the Job Shop Parallel Machine Scheduling (JSPMS) problem. The major objective of the proposed method is minimizing the job completion time and maximizing the job completion rate. Initially, Parallel Colonized Machine Selection algorithm is adapted for selecting optimal machine based on pheromone value. This process also concentrates on minimizing job completion time. Then, a Dynamic Taguchi Machine Scheduling algorithm is applied to the machine selected so that optimal scheduling of operations is said to be performed by applying signal to noise ratio to minimize the computational complexity involved in operation scheduling and enhance the job completion rate. At last, the performance of proposed AC-TPMS method is analyzed using OR Library dataset. The proposed AC-TPMS method reduces the job completion time and computational complexity is 28% and 20% when compared to existing methods respectively. Similarly, the proposed AC-TPMS method enhances the job completion time is 24% when compared to existing methods. The experimental results show that the proposed method is efficient in terms of job completion time, job completion rate and computational complexity when compared with other conventional works

REFERENCES

1. Raviteja Buddala, Siba Sankar Mahapatra, "An integrated approach for scheduling flexible job-shop using teaching-learning-based optimization method", Journal of Industrial Engineering International, Springer, 2019, Volume 15, Number 1, Page(s) 181 - 192
2. Liji Shen, Stephane Dauzere-Peres, Janis S. Neufeld, "Solving the Flexible Job Shop Scheduling Problem with Sequence-Dependent Setup Times", European Journal of Operational Research, Elsevier, March 2018, Volume 265, Issue 2, Pages 503-516

3. Hehua Zhang, Ming Gub, "Modeling job shop scheduling with batches and setup times by timedPetri nets", Mathematical and Computer Modelling, Elsevier, 2009, Volume 49, Issues 1–2, Pages 286-294
4. Jun-qing Li,Quan-ke Pan, Kun Mao, "Hybrid Particle Swarm Optimization for Hybrid FlowshopScheduling Problem with Maintenance Activities", Hindawi Publishing Corporation, The Scientific World Journal, Apr 2014, Volume 2014, Pages 1-11
5. Paweł B. Myszkowski Łukasz P. Olech MaciejLaszczyk Marek E. Skowroński, "Hybrid Differential Evolution and Greedy Algorithm(DEGR) for Solving Multi-Skill Resource-ConstrainedProject Scheduling Problem", Applied Soft Computing, Elsevier, 2018, Volume 62, Pages 1-14
6. Bing Wang, Xiaozhi Wang, Fengming Lan, QuankePan, A Hybrid Local-search Algorithm for Robust Job-shopScheduling under Scenarios", Applied Soft Computing, Elsevier, 2018, Volume 62, Pages 259-271
7. Thitipong Jamrus, Chen-Fu Chien, Mitsuo Gen, and Kanchana Sethanan, "Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Timefor Semiconductor Manufacturing", IEEE Transactions on Semiconductor Manufacturing, 2018,Vol. 31, No. 1, Page(s): 32 - 41
8. Kaizhou Gao, Fajun Yang, MengChu Zhou, Quanke Panand Ponnuthurai Nagaratnam Suganthan, "Flexible Job-Shop Rescheduling for New JobInsertion by Using Discrete Jaya Algorithm", IEEE Transactions on Cybernetics, 2019, Volume: 49, Issue: 5, Page(s): 1944 - 1955
9. Shuaiqun Wang, Aorigeley, Guanjun Liuz, Shangce Gao, "A Hybrid Discrete Imperialist CompetitionAlgorithm for Fuzzy Job-shop Scheduling Problems", IEEE Translations and content mining, May 2016, Pages 1-12
10. Cemal Özgüven, Lale Özbakır, Yasemin Yavuz, "Mathematical models for job-shop scheduling problems with routingand process plan flexibility", Applied Mathematical Modelling, Elsevier, 2010, Volume 34, Issue 6, Pages 1539-1548
11. Imran Ali Chaudhrya and Abid Ali Khanb, "A research survey: review of flexible job shop schedulingTechniques", International Transactions in Operational Research, Wiley, May 2016, Pages 1-41
12. Tianhua Jiang,Chao Zhang,Huiqi Zhu,Guanlong Deng, "Energy-Efficient Scheduling for a Job Shop Using GreyWolf Optimization Algorithm with Double-Searching Mode", Mathematical Problems in Engineering, Oct 2018, Volume 2018, Pages 1-12
13. Pisut Pongchairerks, "A Two-Level Metaheuristic Algorithm forthe Job-Shop Scheduling Problem" Complexity, Mar 2019, Volume 2019, Pages 1-11
14. Raviteja Buddala, Siba Sankar Mahapatra, "Two-stage teaching-learning-based optimization method for flexiblejob-shop scheduling under machine breakdown", The International Journal of Advanced Manufacturing Technology, Springer, Oct 2018, Volume 100, Issues 3, Pages 1419–1432
15. Humyun Fuad Rahman, Ruhul Sarker, Dary Essam, "Multiple-order permutation flow shop schedulingunder process interruptions", The International Journal of Advanced Manufacturing Technology, Springer, May 2018, Volume 97, Pages 2781–2808
16. Ghasem Safari, Ashkan Hafezalkotob, Mohammad Khalilzadeh, "A Nash bargaining model for flow shop scheduling problemunder uncertainty: a case study from tire manufacturing in Iran", The International Journal of Advanced Manufacturing Technology, Springer, Jan 2018, Volume 96, Pages531–546
17. Paolo Renna, "Controllable processing time policies for job shopmanufacturing system", The International Journal of Advanced Manufacturing Technology, Springer, Nov 2012, Volume 67, Pages2127–2136
18. S. C. KIM, P. M. BOBROWSKI, "Impact of sequence-dependent setup time on job shopscheduling performance", Taylor & Francis, International Journal of Production and Research, Jan 2015, Volume 32, Issues 7, Pages 1503-1520
19. B. Naderi, M. Zandieh, S. M. T. Fatemi Ghomi, "Scheduling sequence-dependent setup time job shopswith preventive maintenance", International Journal of Advanced Manufacturing Technology, Springer, Jun 2010, Volume 43, Issues 1, Pages 170-181
20. D. Ahmadkhaniha, M. Heydarzadeh Sohi, A. Zarei-Hanzaki, S.M. Bayazid, M. Saba, "Taguchi optimization of process parameters in friction stir processing of pure Mg", Journal of Magnesium and Alloys, May 2015, Volume 3, Issue 2, Pages 168-172
21. Zhang, S., Li, X., Zhang, B., & Wang, S. "Multi-objective optimization in flexible assembly job shop scheduling using a distributed ant colony system", European Journal of Operational Research, Volume 283, Issue 2, 2020, Pages 441-460
22. Li, K., Chen, J., Fu, H., Jia, Z., & Fu, W. "Uniform parallel machine scheduling with fuzzy processing times under resource consumption constraint" Applied Soft Computing, Volume 82, 2019, Pages 1-13