Available online at www.elixirpublishers.com (Elixir International Journal)

Mechanical Engineering

Elixir Mech. Engg. 96 (2016) 41360-41363

ISSN: 2229-712X

# Optimization of Flow Shop Scheduling Problems using Teaching Learning based Optimization (TLBO)

B.Sravani[1] and K.Nageswara Reddy[2]

[1] Assistant professor, Department of Mechanical Engineering, Sree Vidyanikethan Engineering College( SVEC), Rangampet,Tirupathi.

[2]Department of Mechanical Engineering, Sri Venkateswara University College of Engineering, Tirupati, India.

**ARTICLE INFO**

**Article history:**
Received: 12 May 2016;
Received in revised form: 25 June 2016;
Accepted: 30 June 2016;

**Keywords**
Flow Shop Scheduling,
Taillard's Benchmarks,
TLBO,
Palmers and CDS Heuristic.

**ABSTRACT**

Flow Shop Scheduling is the combinational optimization & NP-hard (i.e. Non-deterministic Polynomial-time hard) problems. In Permutation Flow Shops, the sequence of the jobs is same on all machines. In a Flow Shop Scheduling problem with 'n' jobs that should be processed on 'm' machines. The job can be processed on at most one machine; meanwhile one machine can be processed at most one job. A significant research effort has been committed for sequencing jobs in flow shop to minimizing the make span. Optimization algorithms such as Simulated Annealing (SA), Genetic Algorithm (GA), Ant Colony Optimization (ACO) & Neighborhood Search have played a significant role in solving small scale flow shop scheduling problems. In this paper a recently developed Teaching Learning Based Optimization (TLBO) is proposed method to solve the flow shop scheduling problems to minimize the make span. The proposed algorithm is tested on Taillard Benchmark problems and results are compared with Palmer's & CDS Heuristic methods. The results show that the proposed algorithm is efficient in producing optimal solution and simple, easy to understand.



## 1.Introduction

A flow shop production introduces a manufacturing system where n jobs are processed by m machines in the same order. The permutation flow shop scheduling problem is a well-known scheduling problem that can be formulated as follows: a set of 'n' independent jobs to be processed on a set of 'm' independent machines. Every job requires a fixed processing time on every machine. Each machine can process at most one job at a time and assumed that the jobs are processed by all machines in the same order. The objective of the flow shop scheduling problem is to find an optimal job sequence with minimum make span. The flow shop scheduling problem is usually denoted as Fm/prmu/$C_{max}$ and it is a combinatorial problem with n! Possible sequences.

## 2. Literature Review

In recent works a large number of heuristics and Meta heuristics have been proposed to solve the permutation flow shop scheduling problems, because of complexity developed by exact methods to solve large size problems.

For flow shop scheduling problem, Johnson (1954) proposed algorithm that optimally solves a 2 - machine flow shop problems and for some special cases with 3 machines. Palmer (1965) presented a heuristic to solve the more general m-machine permutation flow shop scheduling problem. This heuristic assigns an index to every job and then produces a sequence after sorting the jobs based on the calculated index.

Campbell, Dudek and Smith (1970) develop another heuristic which is basically an extension of Johnson's algorithm to the m-machine case. The heuristic constructs m-1 schedules by grouping the m original machines into 2 virtual machines and solve the results 2 machine problem by repeatedly using Johnson's.

Regarding Meta heuristics [1] there are different methods for permutation flow shop scheduling problems under different criteria. Genetic Algorithm was proposed by Marcelo Seido Nagano, Rubén Ruiz, and Luiz Antonio Nogueira Lorena [2], Ant Colony optimization Algorithm was proposed by Betul Yagmahan, Mehmet Mutlu Yenisey [2] for solving flow shop scheduling problems. Other methods like Hybrid Scatter Search Algorithm [3], simulated annealing and differential evaluation are proposed to solve flow shop problems.

RV Rao [4] proposed, the TLBO technique used to solve the problem of Job Shop Scheduling. This paper outlines the algorithm implementation and performance when applied to Job Shop Scheduling Problem. The performance measure considered is make span time.

## 3. Teaching learning based optimization (Tlbo)

TLBO algorithm is one of the Nature's inspired population based optimization methods developed by R V Rao [6,7] based on an inspiration from the Teaching - Learning process, which is based on influence of a teacher on the output of learners in a class.

The TLBO algorithm has two phases

1. Teacher Phase
2. Learner Phase

### 3.1 Teacher Phase

During this phase, a teacher tries to bring his or her learners up to his or her level in terms of knowledge. But practically it is impossible and a teacher can only move the mean of a class up to some extent depending on the capability of the class. This follows a random process depends on many factors. Let $M_i$ be the mean at any $i^{th}$ iteration. The teacher will try to move mean $M_i$ towards his or her own level so the new mean will be designated as $M_{new.}$

**Tele:**
**E-mail address: sravani.b.mech@gmail.com**

The solution is updated according to the difference between the existing and new mean given by

Difference Mean = $r_i*(M_{new}-T_F*M_i)$

Where $T_F$ is the Teaching factor which decides the value of mean to be changed and $r_i$ is the random number in the range (0, 1). Value of $T_F$ can be either 1 or 2, which is a heuristic step and decided randomly with equal probability as

$T_F$ = round [1+rand (0, 1) {2-1}]

This difference modifies the existing solution according to the following expression

$X_{new,i} = X_{old,i}$ +Difference Mean

Where $X_{new,i}$ is the updated value of $X_{old,i}$. Accept $X_{new,i}$ if it is gives better function value.

### 3.2 Learner Phase

Learners increase their knowledge by two different means: one through input from the Teacher and the other by self studying. A learner learns something new by self studying than him or her knowledge. Learner modification is expressed as

For any iteration i (i=1: $P_n$)

Randomly select two learners $X_i$ and $X_j$ where $i \neq j$

$X_{new,i} = X_{old,i} + r_i (X_i-X_j)$ if $f(X_i) < f(X_j)$

$X_{new,i} = X_{old,i} + r_i (X_j-X_i)$ if $f(X_i) >= f(X_j)$

Accept $X_{new,i}$ if it gives better function value.

## 4. Mapping of Tlbo Algorithm to Fssp

In the present study of FSSP, the goal is to find the job sequence that minimizes the make span value. The steps of operation are described as follows:

**Step 1:** The initial parameters such as the population size, number of generations, processing time on each machine and the machine sequence are given. Table 2 represents the job processing time on each machine and Table 1 represents the same machine sequence for all jobs. In the solution representation, a solution in FSSP is a job sequence which is represented as a student (x) in TLBO algorithm. Each dimension in a student represents one operation of a job. Each job appears exactly one time in a job sequence. J stands for the operation of job i.

**Table1. An example of job machine sequence for 3-job, 3-machine FSSP.**

| | | |
|---|---|---|
| **M2** | **M1** | **M3** |

**Table 2. An example of job processing time on each machine for5-job, 3-machine FSSP.**

| Job | Machine1 | Machine2 | Machine3 |
|---|---|---|---|
| 1 | 8 | 5 | 4 |
| 2 | 10 | 6 | 9 |
| 3 | 6 | 2 | 8 |
| 4 | 7 | 3 | 6 |
| 5 | 11 | 4 | 5 |

The interpretation of the example is as follows. The job sequence as scanned from left to right, the J3 corresponds to first the job J3 will be processed on all machines, the J2 corresponds to the job J2 will be processed on all machines after J3, the J4 corresponds to the job J4 will be processed on all machines after J2, the J5 corresponds to job J5 will be processed on all machines after J2, the J1 corresponds to job J1 will be processed on all machines after J5. Thus, the feasible schedule can be constructed as shown in Fig-2.

| | | | | |
|---|---|---|---|---|
| J3 | J2 | J4 | J5 | J1 |

**Fig1. Example of job sequence representation for 5×3 FSSP.**

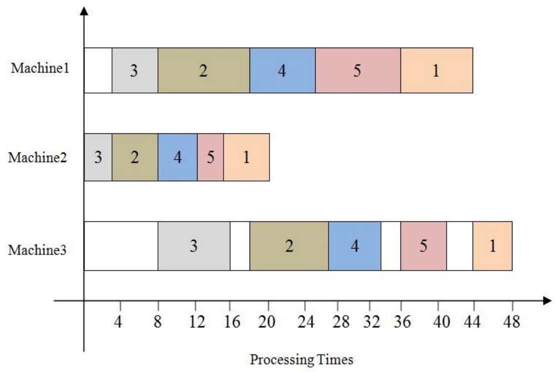


**Fig 2. Feasible schedule constructed from the job sequence in Fig. 1**

Step 2: Now the mean of the make span (M) is calculated and any one solution is selected which is nearer to the mean ($M_D$). The best solution will act as the teacher for that iteration ($M_{new}$).

The teacher ($X_t$) tries to shift the mean from M towards $X_t$ which will act as a new mean for the iteration. The difference solution ($D_D$) is updated by old mean solution ($M_D$) and new mean solution ($M_{new}$) using Position Based Crossover (PBX) mechanism. An example of PBX is shown in Fig. 3

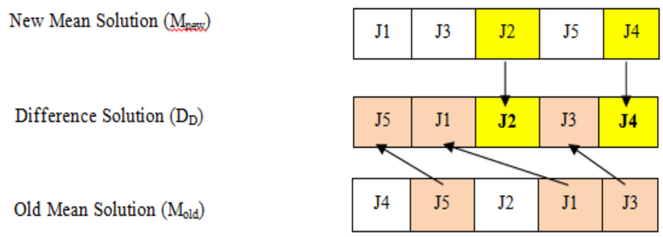


**Fig 3. updating difference solution using new mean and old mean based on PBX method.**

Based on the PBX method, a set of job operations from the new mean is selected randomly. Each dimension in the job sequence of the new mean is selected to produce the new difference. The jobs already selected from the new mean are ignored and are not selected again from the old mean. The job operations on the old mean that are not yet selected from the new mean are selected and placed into empty positions from the left to the right of the job sequence in the difference solution. To guarantee that each job is included exactly one time in the difference solution, if any job has already been selected (from either the old or the new mean), it is skipped and the next job is considered.

Then the current solutions are updated by using the relation shown below

$X_{new} = X_{old} + D_D$

The obtained difference is used to the current solution to update its values using Position Based Crossover mechanism. By considering difference ($D_D$) as the new mean ($X_{new}$) and each solution in the population as old mean ($X_{old}$), one at a time, TLBO updates the solutions in the teacher phase.

Step 3: The solutions (Xnew) in the teacher phase are improved in the learner phase. In the learner phase it is assumed that the students improve their knowledge by self studying instead of mutual discussion as was proposed in Rao [4] .This modification is applied to reduce the complexity in solving the FSSP using TLBO. This self studying concept is applied by using the variable neighborhood method because in this method the solutions are improved by itself without depending upon the neighboring solutions or any other solutions which are better than it.

In this way the solutions in the teacher phase are updated by solutions in the learner phase using variable neighborhood search mechanism.

A local search based on the Variable Neighboring Search method (VNS) [5] is performed on the teacher's solutions to improve the solution quality by itself, it sometimes takes a long time to reach better solutions while solving large scale flow Shop Scheduling. To overcome this teacher phase helps VNS to find solutions as early as possible.

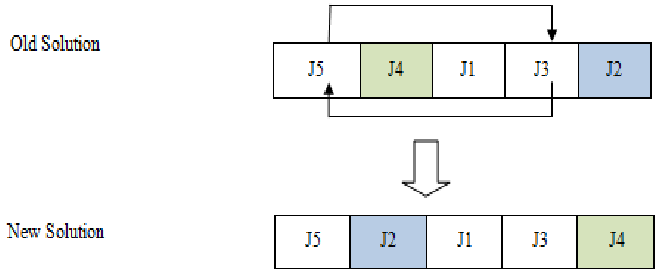


**Fig 4. Exchanging process in VNS method for new solution.**

i and j are the random integer numbers between 1 and n, Exchanging Process (x, a, b) means exchanging the job operations in solution x between i[th] and j[th] dimensions, i≠j. Inserting Process (x, a, b) means removing the job operation in solution x from the ith dimension and inserting it in the j[th] dimension. The example of the exchanging process and the inserting process are shown in Figs. 4 and 5 respectively.

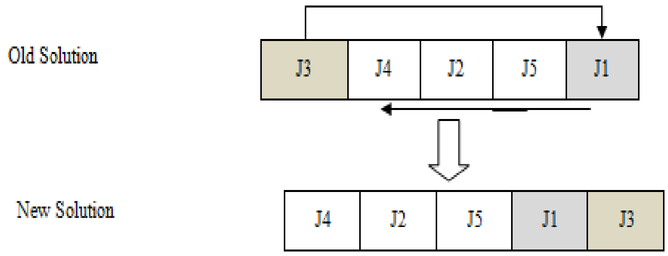


**Fig 5. Inserting process in VNS method for new solution.**

**Step 4**: If the termination criterion is satisfied then the algorithm stops, else it goes to the next run or Iteration.

### 5. Flow Shop Scheduling

In flow shop scheduling problem, there are n jobs, each requires processing on m different machines. The process sequence for all the jobs is same. But the processing times for various jobs on a machine may differ. If an operation is absent in a job, the processing time of the operation of that job is assumed as zero.

A mathematical model to solve flow shop scheduling problem as follows [10]:

Minimize: $C_{max}$

Subject to:

$C_{max} \geq C_{im}$ for all i,

$C_{ij} = S_{ij} + PT_{ij}$ for all i and j,

$S_{ij} \geq R_j$ for all i,

$C_{ij} \geq C_{i,j-1} + PTij$ for all i,

$Cij \geq 0$ *for all i, j.*

Where

n = Number of jobs

m = Number of machines

j = Index for jobs

i = Index for machines

$PT_{ij}$ = Processing time of job j on machine i

$C_{ij}$ = Completion time of job i on machine j

$C_{im}$ = Completion time of job i on machine m

$C_{max}$= Make span value

C* = Optimal make span value

$R_j$ = Ready time of job j

$S_{ij}$ = Starting time of job i on machine j

### 6. Experimental Results

The present study aims to solve the flow shop scheduling problems, to evaluate the performance of the TLBO algorithm (i.e. to evaluate the solution quality of the proposed algorithm). The performance of the proposed TLBO algorithm is evaluated by testing on 15 Taillard's benchmark problems. The size of these problems ranges from 6 to 20 jobs and 5 to 20 machines. To solve the above benchmark problems, population size is set to 25 and no. of iterations is set to 100, so that the total function evaluation becomes 5000.Each problem had been solved 10 times with different random initial solutions and constant function evaluations. The algorithm TLBO is coded in MATLAB R2014a version and ran on a PC with a 3 GHz Pentium® Dual Core CPU and 2GB of RAM Memory.

### 7. Conclusion

Flow Shop Scheduling Problems (FSSP) can be solved by Palmers, CDS and TLBO. In this paper TLBO is applied to FSSP to obtain optimal sequence with best make span. From Experimental results it can be said that TLBO gives better solution in terms of best make span than Palmers & CDS.

**Table 3. TLBO, Palmers & CDS results comparison on the Taillard's benchmark problems.**

| Test Instance | Problem Size | | Palmer's Solution | CDS Solution | TLBO Solution |
|---|---|---|---|---|---|
| | Jobs | Machines | | | |
| Taillard001 | 20 | 5 | 1384 | 1398 | 1278 |
| Taillard002 | 20 | 5 | 1439 | 1424 | 1359 |
| Taillard003 | 20 | 5 | 1162 | 1249 | 1081 |
| Taillard004 | 20 | 5 | 1420 | 1418 | 1293 |
| Taillard005 | 20 | 5 | 1360 | 1323 | 1236 |
| Taillard011 | 20 | 10 | 1790 | 1757 | 1590 |
| Taillard012 | 20 | 10 | 1948 | 1854 | 1660 |
| Taillard013 | 20 | 10 | 1729 | 1645 | 1509 |
| Taillard014 | 20 | 10 | 1585 | 1547 | 1386 |
| Taillard015 | 20 | 10 | 1648 | 1558 | 1424 |
| Taillard021 | 20 | 20 | 2818 | 2579 | 2316 |
| Taillard022 | 20 | 20 | 2331 | 2285 | 2116 |
| Taillard023 | 20 | 20 | 2678 | 2565 | 2349 |
| Taillard024 | 20 | 20 | 2629 | 2434 | 2236 |
| Taillard025 | 20 | 20 | 2704 | 2506 | 2308 |

The TLBO method is compared with Palmer and CDS Heuristic and results shown in table 3, t is found that obtained make span value by TLBO is better than the CDS & palmers Heuristic.

It can be said that the TLBO algorithm can be effectively used for FSSP.TLBO method is simple and easy to understand. The TLBO can be applied to other Scheduling Problems such as Batch Scheduling, FMS Scheduling and AGV Scheduling etc.

**References**

1) K.Mallikarjuna, Venkatesh.G, Somanath.B, A Review on Job Shop Scheduling by Non Conventional Optimization Algorithm, International journal of engineering Research and applications Vol. 4, Issue 3 (Version 4), March 2014,pp.11-19.

2) Betul Yagmahan, Mehmet Mutlu Yenisey, multi objective flow shop scheduling Problem by Ant colony optimization, Computers & Industrial Engineering 54 (2008) 411–420.

3) R. Hariharan and R.J. Golden Renjith Nimal, Solving Flow Shop Scheduling Problems by Hybrid Genetic Scatter Search Algorithm, Middle East Journal of Scientific Research 20 (3): 328-333, 2014.

4) H. S. Keesari & R. V. Rao, Optimization of the job shop scheduling problems using teaching-learning-based optimization algorithm, OPSEARCH (Oct–Dec 2014) 51(4):545–561.

5) Pierre Hansen, Nenad Mladenovic, José A. Variable neighborhood search- application & methods (2008) 6:319–360.