International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)
ISSN (P): 2249-6890; ISSN (E): 2249-8001
Vol. 8, Issue 2, Apr 2018, 751-774

SHUFFLED FROG LEAPING ALGORITHM FOR SOLVING OPTIMIZATION OF TOTAL HOLDING COST IN JOB SHOP SCHEDULING PROBLEMS

G. RAMYA[1], M. CHANDRASEKARAN[2] & A. PARTHIBAN[3]

[1]*Department of Mechanical Engineering, Rajalakshmi Engineering College, Thandalam, Chennai, India*

[2,3]*Department of Mechanical Engineering, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, India*

ABSTRACT

The manufacturing System is enabled with an excellent knowledge of the production plan and proper scheduling of machinery process. Challenging combinatorial optimization problems are encountered even in the job shop scheduling problems. Heuristics algorithms are developed in a scholastic search way in which natural big buoyancy is maintained. These are developed to bring optimized results in stipulated time with respect to optimally schedule. This article deals with minimizing the total holding cost of completed and in-process products with consideration of tardy jobs and without consideration of tardy jobs with Shuffled Frog Leaping Algorithm (SFL). Applying SFL algorithm to minimize the total holding cost which is the sum of earliness finished product inventory holding cost, intermediate inventory holding costs, work in process and tardiness. Several benchmark problems of different sizes, which are commonly used for Job Shop Scheduling Problems of minimizing the total holding costs and makespan are produced. The Results are compared with literature results in terms of total holding cost, stipulated time and computational time the Shuffled Frog Leaping algorithm performs result oriented than other Heuristics Algorithm.

KEYWORDS: Heuristic Algorithm, Job Shop Scheduling, Shuffled Frog Leaping Algorithm, Tardy Jobs & Total Holding Cost

Received: Feb 20, 2018; **Accepted:** Mar 02, 2018; **Published:** Mar 14, 2018; **Paper Id.:** IJMPERDAPR201887

Original Article

I. INTRODUCTION

A schedule is an allocation of tasks to the time intervals on the machines. The aim is to find a schedule that minimizes the overall completion time, which is called the makespan. In the job shop scheduling, problem n jobs have to be processed on m different machines. Each job consists of a sequence of tasks that have to be processed during an uninterrupted time period of a fixed length on a given machine. Due dates are treated as deadlines and require the job-shop scheduling to meet specific due dates in order to avoid delay penalties, including customer's bad impression, cost of lost future sales and rush shipping cost.

Efficient Methods are traditional approaches consider technological advances in both processes and equipment as the key to success and the right way to remain competitive. Many valid approaches and its advances are compared and shared between competitors in rapid form. Each and every approach have its own valid solution exclusively [1].

Akers proposed a graphical approach for the 2 X m problem [2]. Hefetz et al. have developed an efficient approach for then X 2 problem where all operations are of unit processing time [3]. Williamson et al. proved that the feasible schedule determining with a makespan can be done in polynomial time [4].

The Job shop scheduling problem has been described as NP which means Non deterministic Polynomial time. Lenstra et al [5-6] solved the 3 X 3 problem, then X 2 instance with no more than 3 operations per job and then X 3 problem with no more than 2 operations per job. Lenstra et al. proved that then X 2 instance in which operations last for no more than 2 units of processing time and then X 3 problem in which all operations are of unit processing time belongs to the set of NP instances [7]. Mattfeld et a proposed randomly generated solutions with precedence relations which are not uniformly distributed [8]. A 10 X 10 problem proposed by Fisher et al. was solved by Carlier et al. [9]. Shmoys et al. proposed several poly-logarithmic approximations for evaluating an optimal schedule with makespan minimization criteria [10]. French predicted that no efficient algorithms will ever be developed for the majority of scheduling problems [11]. As a result, the focus of optimization research has turned to be enumerative approaches.

Branch and Bound (B&B) algorithms use a dynamically constructed tree structure as a means of representing the solution space of all feasible sequences. The branching procedure replaces an original problem by a set of new problems that are mutually exclusive and exhaustive subproblems, partially solved versions and smaller problems than the original. The two most common branching strategies were Generating Active Schedules (GAS) and Settling Essential Conflicts (SEC) [12]. Barcker et al. have generated Lower bounds by reducing job shop scheduling problem into subproblems of dimensionality [13]. The B&B search technique was initially studied by Brooks et al. [14]. Using a dynamically constructed tree structure represents the solution space of all feasible sequences Each node Search begins at the topmost level and a complete selection is achieved only at the lowest level has and (p) level in the search tree represents a sequence of operations partially. The branching operation sequences determine the next set of possible level nodes from sequence search progress [15].

Approximation procedures applied to a job shop problem were first developed on the basis of priority dispatching rules and due to their ease of implementation and substantially reduced computational requirement. Some of the Priority Dispatching Rules are Earliest Due Date (EDD), Shortest Processing Time (SPT), Minimum Slack, First (MSF), First come, first served (FCFS), Most work Remaining (MWR), Most Operations Remaining (MOR), Least Work Remaining (LWR), Random Analysis (RA). The earliest work on PDRs was done by Jackson [16]. The comprehensive survey of scheduling PDRs heuristics was done by Panwalker et al. where 113 PDRs were presented, reviewed and classified [17]. Haupt et al. provided an extended discussion and summary of these and many other PDRs [18]. The comparative study was given by Chang et al. who evaluated the performance of 42 PDRs using a linear programming model [19]. Lawrence compared the performance of ten individual priority dispatch rules with a randomized combination of these rules [20]. The results found from PDRs had more deviations from optimum. The results suggested that PDRs are more suitable as an initial solution technique rather than being considered as a complete JSP.

It has been recognized by many researchers that scheduling problems can be solved optimally using mathematical programming techniques and one of the most common forms of mathematical formulation for the job shop scheduling problem was the Mixed Integer Linear Programming (MIP) format of Manne [21]. Blazewicz et al. Emphasized the difficulties of JSP and indicated that mathematical programming models have not been achieved enough breakthroughs for scheduling problems [22]. Any success that has been achieved using mathematical formulations can be attributed to Lagrangian Relaxation (LR) approaches and decomposition methods Mathematical formulations were combined with other techniques and applied in the calculation of the lower bound. Results indicate that the lower bound generated was not optimal and it is difficult to calculate and takes excessive computing time. It is evident that mathematical approaches are inadequate for thejob shop scheduling problem. Consequently, the main focus of enumerative approaches for the job-shop is a branch and bound techniques.

In earlier research, the job-shop scheduling problem has been extensively studied with the objective of minimizing some functions of the completion times of jobs. Several techniques have been proposed and different heuristics have been designed and developed for solving the minimum makespan problem, the minimum total tardiness problem and so on. SFHM algorithm was used for minimizing mean tardiness and mean flow time multiobjective criteria [23]. An effective SFLA was used for minimizing maximum completion time (i.e., makespan) [24]. In this work, SFLA and SFHM algorithm are used for solving the scheduling problem to meet due dates in a simple job shop. It is developed to approximately minimize the total holding cost which corresponds to the sum of product inventory cost and in-process inventory cost. Several benchmark problems are solved by the proposed algorithms and the results are compared with literature results.

II. JOB SHOP SCHEDULING

A. Scheduling Objectives

The scheduling is carried out to meet various objectives. These objectives are decided upon the situation, market demands, and the customer’s satisfaction. The objectives considered under the time and cost minimization are listed in Table 1.

Table 1: Main Objectives of Scheduling

Sl. No	Time Minimization	Cost Minimization
1	Minimize machine idle time	Minimize the costs due to not meeting the due dates
2	Minimize the mean flow time	Minimize the maximum lateness of any job
3	Minimize the mean tardiness	Minimize the total holding cost with no tardy jobs
4	Finish each job as soon as possible	Minimize the total holding cost with tardy jobs
5	Finish the last job as soon as possible	Minimize the number of late jobs

B. Job Shop Scheduling Problem

Typical scheduling problems involve minimizing the maximum gj(t) value (the maximum cost problem) or minimizing the sum of gj(t) values (the total cost problem). Scheduling is defined as the art of assigning resources to tasks in order to ensure the termination of these tasks in a reasonable amount of time (25). The term ‘Scheduling’ in manufacturing systems is used for the determination of the sequence of operations in which parts are to be processed over the production stages. To meet an optimal objective solution or set of objectives these approaches are used for the determination of the starting time and finishing time of processing of each part. Some other cases scheduling, problem is addressed after the orders are released into the shop floor, along with their process plans and machine routings [26]. Scheduling plays a crucial role to increase the efficiency and productivity of the manufacturing system. The problem of scheduling is one of the operational issues to be addressed in the system on a daily or weekly basis. Job shop scheduling problems are Non-Polynomial (NP) hard] so it is difficult to find optimal solutions [27].

C. Single Machine Scheduling

In the single machine scheduling, problem n jobs have to be processed with a single operation. The concept of single machine scheduling consists certain conditions. At time zero a set of independent single operation jobs are available for processed, A set-up time of each job is independent, Job description are well known in advance, No machine kept idle when work is processing and each job is processed till its completion without a break. The basic data are necessary to describe jobs in a deterministic problem mentioned in the following Table.2.

Table 2: Basic Data's for Deterministic Problem

Sl. No.	Basic Data & Representation	Description
1	Processing Time (t_j)	Time required to process job J. It will include both actual processing time and set-up time.
2	Ready Time (R_j)	Time at which job j is available for processing. It is the difference between arrival time and the time at which the job is processing.
3	Due Date (d_j)	Time at which the job j is to be completed.
4	Completion Time (C_j)	Time at which the job j is to be completed in sequence.
5	Flow Time (F_j)	Amount of time job j spends in the system ($F_j = C_j - r_j$)
6	Lateness (L_j)	Amount of time by which the completion time of job j differs from the due date ($L_j = C_j - d_j$)
7	Tardiness (t_j)	Lateness of job j if it fails to meet its due date, or zero $T_j = \max\{0, C_j - d_j\}$

D. Flow Shop Scheduling

In the Flow shop scheduling, problem n jobs, each job has to be processed on m different machines. The concept of single machine scheduling consists certain conditions. At time zero a set of multiple operation jobs are available for processing, A set-up times for the sequence operations are independent, Job description is well known in advance, m different machines are available for continuous processing and each individual operation of jobs are processed till its completion without a break.

E. Job Shop Scheduling

In the job shop scheduling, problem n jobs have to be processed on m different machines. Each job consists of a sequence of tasks that have to be processed during an uninterrupted time period of a fixed length on a given machine. So the maximum of completion time needed for processing all jobs is subjected to the constraints that each job has a specified processing order through the machines and that each machine can process at most one job at a time.

F. NP-Hard Problems

Scheduling problem is NP-hard because it is little consolation for the algorithm designer who needs to solve the problem. By despite a theoretical evaluation all NP–hard problems are not equally hard from a practical evaluation. The NP-hard problems can be solved pseudo polynomial using dynamic programming. The sizes of the problems are not large enough to provide satisfactory results in heuristic approach.

G. Pseudo Polynomial Concept

Polynomial time concept exists for some NP-hard problems under the appropriate encoding of the problem data. Such problems are referred to as NP-hard in the ordinary sense and the algorithms are called pseudo polynomial Problem P is called strongly NP-hard if a pseudopolynomial concept for it does not exist. Algorithms which are Polynomial or Pseudo polynomial are applicable for single machine scheduling. Lawler (1973) developed Lawler's algorithm which constructs a sequence in a reverse to get an optimum sequence. The following criteria's are considered to get optimum solutions when the algorithms are pseudo polynomial [27]. Some existing criteria's are minimax criteria, Maximum lateness & related criteria, earliest due date criteria, Total weighted completion time criteria, Optimum sequence criteria, weighted number of late jobs and Total weighted tardiness.

III. MINIMIZATION OF TOTAL HOLDING COST

A. Total Holding Cost

Normally, manufacturing system consists active period starts from the first day of production on the machine with a certain set of actions and operations. In general Meeting, the due dates are the most important goal of scheduling to avoid the delay penalties, including customer's bad impression, lost future sales. Due-date oriented functions, whereas the main aim of optimizing the makespan is to minimize the total holding costs and maximizing the output.

During the production and process time the total holding cost is considered the raw materials holding cost, running process holding cost with stipulated time, labors cost based on the number of employees, machine cost may be owned machine or hired machine, power supply cost based on electricity utilization, Fuel cost, Transportation cost, stock holding cost and inventory management cost. Total accountability on every unit should me readily available in every set of job production. The total holding cost in every production should me equal to + or – 1 deviate from actual. When the production function starts the process management can be completed or to be completed with regular breakeven analysis by applying optimization methods to meet the regular benchmark of production cost management. Proper scheduling of machining processes and operations are enabled in master production schedule in which production to be reached on climax within the stipulated time with an excellent knowledge of production engineer. The employee to avoid absenteeism is to bring out per capita per month analysis. One labor produce per day products worth of + or – k thousands. 16 x 3 Labors produces per day products worth 16 x 3 x 7 is equal to 336 k thousands (3.36 + or – L per day).

B. Minimization of Tardiness Cost

A tardy job is the sum of the rejected part cost and penalty cost. To minimize the total holding cost consider the minimization of work in the process holding cost with running holding cost, minimization of finished product inventory holding cost and minimization of the number of tardiness.

C. Heuristics Algorithms

The heuristic approaches were also employed for various engineering application problems due to their robustness and convergence to global optima. Heuristic method of learning involves discovery and problem solving using reasoning and past experience. An approach without formal guarantee of performance can be considered a “heuristic”. These heuristic approaches are used in a practical situation when no better methods are available. The following section deals with the various heuristic approaches like Artificial Intelligence, Bottleneck based heuristics, Local search approaches, Meta Heuristics and Hybrid Approaches in earlier research work.

Heuristics such as Tabu search, Hybrid shuffled frog leaping, Branch and Bound Technique, Bee colony optimization, Scatter search, Hybrid Metaheuristics, Shifting Bottleneck procedure, Shuffled Leaf Frog Algorithm, Ant colony optimization, and Greedy Randomized adaptive search. Such performance measures give us some insight into the reliability of a particular procedure.

D. Shuffled Frog Leaping Algorithm

Eusuff et al. [28] proposed a new meta-heuristic algorithm called Shuffled Frog Leaping Algorithm for solving scheduling problems with discrete decision variables. SFLA is a population-based cooperative search metaphor combining the benefits of the genetic-based memetic algorithm and the social behavior based particle swarm optimization Inspired by natural memetics [29]. Muzaffar Eusuff and Lansey [30] described the algorithm is imitating the total sequence of modeling process and searching for best food with behavior of frogs placed on separately positioned stones in a pond and also SFLA has been tested with a large number of combinatorial problems and found to be efficient in finding global solutions [31-32]. The SFLA is a population-based cooperative search metaphor inspired by natural memetics and consists of a frog leaping rule for local search and a memetic shuffling rule for global information exchange.

The SFLA comprises a set of an interacting virtual population of frogs partitioned into different groups population memeplexes which are referred to searching for food [33]. The algorithm functions are simultaneously independent in local search of each memeplex [34]. In terms of processing time and makespan, the SFLA compares the results rapid favorably with the Sheep Flock Heredity Model Algorithm, Artificial Intelligence System, Genetic Algorithm, and Particle Swarm Optimization [29, 35, and 36].

Mohammadreza Farahani et al. [29] identified a new hybrid algorithm called Hybrid Shuffled Leaping Frog Algorithm based on the identification of the weaknesses of the basic SFLA. At First, the SFLA is initially applied to different functions and to identify the fundamental weaknesses of this method as per the elimination of the effective frogs from memeplexes by solving procedure in consequence order. This method is similar to the SFLA, partitions particles into different groups called memeplexes and identified the best particle in each memeplex thereafter determines its movement through the search space in each iteration of the algorithm toward the global best particle and the worst particle in each memeplex keeps track of its coordinates in the solution space by moving toward the local best particle in the same memeplex.

Chen Fang et al.[37] proposed to encode concept for the virtual frog as the extended activity list and decode it by the SFLA-specific serial schedule generation scheme. The initial populations are identified and generated by the mutual based shuffling method and the priority rules. The large group of populated virtual frogs is separated into many sets of memeplexes is the next stage and by applying the effective resource- based planning each memeplex are evolves the crossover. Combining the permutation-based local search and forward-backward improvement is to enhance the improved exploitation ability. Virtual frogs are periodically shuffled and rearranged into new set off memeplexes are maintained by the diversity of each memeplex. HSLFA also has a distinct advantage over the SFLA in that it reduces the probability of the particles being trapped in the local minima by directing the best local particle toward the global best particle [34].

E. Drawbacks of Direct Approaches for Job Shop Scheduling

Job shop scheduling problems are NP-hard so that many direct approaches are not performing well due to more complexity of the problems. Many researchers are identified and observed valid drawbacks indirect approaches. Direct approaches are giving optimal solutions to very small size problems. In multi -objective optimization, direct approaches are not efficient. Indirect approaches, the convergence to an optimal solution depends on the chosen initial random solution and the results tend to stick with local optima. These techniques start with a single point and its follow a deterministic rule. Direct approaches are not efficient when practical search space is too large. Branch and Bound approach, Priority Dispatching Rules are giving solutions which have more deviation from optimal solutions. The above drawbacks stress the researchers to search, develop the efficient heuristic approaches.

F. Trends of Heuristic approaches for Job shop Scheduling

The heuristic approaches have more benefits compared with the direct approaches. The heuristic approaches produce the optimal solution for various size problems. Objective functions have given more importance than derivatives. Many Heuristic approaches use a population of points during the search. Initial populations are generated randomly which enable to explore the search space is large. These approaches efficiently explore the new combinations with the available knowledge to find a new generation.

Though an extreme work has been on solving JSS optimization using Metaheuristics and local search techniques are still the major potential area yet to be explored by the researchers using Efficient Heuristics Algorithms

IV. PROBLEM DESCRIPTION

A. Job Shop Scheduling Problem for Minimizing Total Holding Cost Subject to Consideration of Tardy Jobs

In-process cost is the sum of the running process cost, machining cost, raw material storage cost running cost, lateness, Earliness, Absolute deviation, squared deviation, unit penalty cost. Inventory holding cost is the sum of the work in the process holding and finished goods holding cost. A tardy job is the sum of the rejected part cost and penalty cost. To minimize the total holding cost consider the minimization of work in the process holding cost with running holding cost, minimization of finished product inventory holding cost and minimization of the number of tardiness.

Formulation of Objective Functions

Consider a non-preemptive job shop with m machines (M_i = i, …. m) & n jobs (N_i= i, …n). When j_i is the set of a job to be processed on machine M_i. The operation sequence of the job j are denoted by $O_{ii,}$ where i^{th} operation on j^{th} machines M_j. Objective functions depend on due date which is associated with the jobs. A job (j_i) consists of the number of (n_i) operations (O_{i1}, O_{i2},….O_{in}). The operation O_{ij} is a processing requirement P_{ij} and may be processed on any of machines in M_{ij} (set of machines) with ready time R_j. If each job, j_i consists only of one operation ($n_i = 1$) then the identity of job j_I with operation O_{i1} & the processing requirement $P_{i.}$ In each job, k has a constant weight (cost). $w_k > 0$. In every problem of due date (d_k) consideration have almost different due dates which are NP complex, Where k=1,2,..n. When $C_k > d_k$, the decision maker incur the penalty function of $W_{k.}$ If the decision makers incurs no penalty for the k^{th} job is no tardy jobs (when C_k less than or equal to $d_{k.}$ This case is formulated as multiple resource operations and schedule allocation problem with computation variables, which can be solved to optimally in polynomial and stipulated time with makespan. All the jobs (j=1,2,..n) must be sequenced before processing of the jobs begins. The main objective is to optimize the number of job sequences with stipulated time S_k is j_1, j_2, …. j_n. All the jobs to be completed after the due dates end with makespan, which minimizes the expected weighted number of tardy jobs T_{ij}. The cost function c_{jj}, which measures the cost of completing j_i at the time (t), due date (d_i), Earliness cost (E_j) and defining function f_i.

Total holding Cost (h_{ij}) = (in process cost + inventory cost) with tardy jobs cost

$$\min f = \sum_{k=1}^{P} \left\{ \sum_{j=1}^{n}\sum_{j=1}^{m} [t_{ij}^{(k)}, c_{ij}^{(k)}]^{x_{ij}} + [t_{ij}^{(k)}, d_{ij}^{(k)}]^{x_{ij}} + [t_{ij}^{(k)}, I_{ij}^{(k)}]^{x_{ij}} + [R_i]^{x_{ij}} + [E_j, W_{ij}, c_{ij}]^{x_{ij}} \right\} \quad \text{----------- (1)}$$

Subject to

$x_{ij} = \{1,0$

$d_{ij}^{(k)} = i < k < p$

$I_{ij}^{(k)} = 0 \quad < k < 1$

$t_{ij} < 1$ unit

For a given schedule (S), c_{ij} is the time at which job j finished processing on machine i and w_{ij} is the weighted time of job j spends in the queue before the first machine i. Already times, processing times and due dates are assumed to be integer. In the above function k^{th} job is performed in an i^{th} machine with j^{th} operation with unit time consideration for time t_i and cost $j_{cost.}$ If the i^{th} machine is assigned with j^{th} operation for the first job is $X_{ij}^{(1)}$ is 1,0. If the i^{th} machine is assigned with j^{th} operation for the k^{th} job is $P_{ij}^{(k)}$ is 1,0. Sub -objective functions are also be formulated to minimize the tardiness cost t_j, Earliness cost $E_{j,}$ finished product holding cost h_{ij} &w_{ij} and in-process holding cost $P_{ij.}$ [38]

Consideration of Separate Objective Functions

Minimization of Tardiness cost function

$$C(s) = \sum_{i=1}^{n} \pi_i t_i \tag{2}$$

Minimization of Earliness cost

$$C(s) = \sum_{j=1}^{n} €_j E_j \tag{3}$$

Minimization of product holding cost

$$C(s) = \sum_{j=1}^{n} \sum_{i=2}^{m} h_{ij} w_i \tag{4}$$

Minimization of in process holding cost

$$C(s) = \sum_{j=1}^{n} h_{ij} (c_{ij} - P_{ij} - R_{ij}) \tag{5}$$

The first job cost is algebraic sum of Inventory holding cost ($I_{ij}^{(k)}$), Processing cost ($t_{ij}^{(k)}$).

For Job (1) = $I_{ij}^{(1)} + P_{ij}^{(1)} + t_{ij}^{(1)}$ where $I_{ij}^{(1)}$ is zero

For Job (2) = $I_{ij}^{(2)} + P_{ij}^{(2)} + t_{ij}^{(2)}$ where $I_{ij}^{(2)}$ is x

For Job (k) = $I_{ij}^{(k)} + P_{ij}^{(k)} + t_{ij}^{(k)}$

The sequence of jobs $J_{ij}^{(k)}$ have due dates to minimize the total holding cost with the number of tardy jobs to find the sequence of jobs. Further, For Solving the above objective functions to find an optimum solution Heuristics method named shuffled frog leaping algorithm has to be implemented and validated.

B. Job Shop Scheduling Problem for Minimizing Total Holding Cost Subject to no Tardy Jobs [39]

A set of I jobs has to be processed on K machines. The processing of job J_i (i=1, 2,…,I) on a machine is called an operation and each operation can be performed by only one machine. The processing order of a job is given. Let $O^l{}_i$ (l=1, 2, …, L_i) denote the l^{th} operation of job J_i, where L_i corresponds to the number of operations for job J_i. The processing time p^l_i of operation $O^l{}_i$ is pre-specified. Each machine k (k=1, 2,…., K) can process only one operation at a time. Pre-emption is not allowed, and each job is available for processing at time 0. The due date d_i of job J_i is pre-specified by the associated customer. Every job must be completed before or just on its due date and no tardy jobs are allowed. The holding cost in the shop floor incurs the in-process time once a job begins processing, and if a job is completed earlier than its due date, then earliness cost can be induced in the shop floor to holds the finished job. Assume $w^{l-1}{}_i < w^l{}_i$, where $w^l{}_i$ (l=1, 2, …, L_{i-1}) denotes the holding cost per unit time for in-process product in idle time from end of operation $O^l{}_i$ to start of operation $O^{l+1}{}_i$, and $w^{Li}{}_i$ denotes the holding cost per unit time for completed product from end of operation $O^{Li}{}_i$ to due date d_i. This assumption means that holding cost for in-process product is increasing based on the progress of the operation. Let $C^m{}_i$ (decision variable) denote the completion time of operation $O^m{}_i$ and E_k the set of operations to be performed on machine k, then the problem is as follows:

The objective function corresponds to the minimum total weighted flow time from the determined starting time to the pre-specified due date for every job. Eqs. (2) and (3) are the conjunctive and disjunctive constraints, respectively. Eq. (4) is the due-date constraint, and Eq. (5) implies that each job is available for processing at time 0.

$$\text{Minimize } f = \sum_{i=1}^{1} \Big\{ \sum_{l=1}^{L_i-1} w^l_i \,(C^{l+1}_i - p^{l+1}_i - C^l_i) + w^{L_i}_i (d_i - C^{L_i}_i) \Big\} \qquad \text{........ (1)}$$

Subject to

$$C^m_i - C^{m-1}_i \geq p^m_i,\ \ i = 1,\ldots, I, m = 2,\ldots,L_i, \qquad \text{........ (2)}$$

$$C^m_i - C^n_i \geq p^m_i \ \vee\ C^n_j - C^m_j \geq p^n_j, \qquad \text{........ (3)}$$

$$d_i - C^{L_i}_i \geq 0,\ i = 1,\ldots, I \qquad \text{........ (4)}$$

$$C^l_i - p^l_i \geq 0,\ i = 1,\ldots, I \qquad \text{........ (5)}$$

V. PROPOSED METHODOLOGIES

A. Shuffled Frog Leaping Algorithm

In this section, an SFLA for solving the JSS problem with minimizing total holding cost and makespan criterion are proposed by population initialization, partitioning scheme, memetic evolution process, shuffling process, and a local search. SFLA was a combination of memetic Algorithm and Particle Swarm Optimization. It has been performed from the memetic evolution of a group of frogs when seeking for food. The initial population of frogs was partitioned into groups or subsets called "memeplexes" and the number of frogs in each subset was equal.

The SFLA was followed two search techniques a) local search and b) global information exchange. Based on local search to reach the makespan, the frogs in each subset improve their positions to have more foods. After a local search, obtained information based on Global information exchange between each subset was compared to other to produce best sequence way of schedule. Each operation is decided by meeting pre-specified due dates and minimizing an objective function. An initial population of the sequence generated randomly by increasing order and selected sequence divided into a number of memeplexes.

Local Search Procedure

The division is done with the high level frog (column sequence) arranged in the first memplex, the second one arranged in the second memplex, the last frog to the last memplex and repeated frog back to the next order memplex. Fitness function evaluated within the limits that the memplex is infeasible.

Global Information Exchange

The best frog memplex values were identified each subset was compared to each other to produce best sequence way of schedule. For each iteration the frogs with the best fitness and worst fitness were identified and also the frog with the makespan schedule was identified. Finally, if the convergence criteria is not satisfied the position of the worst frog for the memplex is adjusted and new subsets of memplex will be created for the next iteration.

B. SFLA Heuristics Algorithm Procedure

Start;

Step 1: Randomly generate the population size of frogs P in Feasible situation &

Initialize the population size equal to no. of memeplexes;

Step 2: For each individual population P, calculate the fitness size (i) & Calculate the size of each memeplex subsets ;

Step 3: Rearrange the population size randomly;

Step4: Evaluate P based on the hierarchy order of their fitness; & Divide P into m memeplexes with i=1 to no. of generations;

Step 5: Perform a Local search to Improve frog position to have the best food;

Step 6: For each memeplex; Determine the best and worst frogs; Improve the worst frog position by removing worst frogs in a frame;

Step 7: Shuffle Each improved memeplexes and Combine the evolved

memeplexes;

Step 8: Sort the population P in descending order of their fitness;

Step 9: If Convergence criteria satisfied (Make pan) move to end or else move to step 1

End;

C. Representation of Solution Seed (Sequences) in Job Shop Simulator

Consider the five-job five-machine problem as shown in Table 3. and Table 4. Suppose a seed sequence is given as [5, 4, 3, 2, 1], where 1 stands for job j1, 2 for job j2, 3 for job j3, 4 for job j4 and 5 for job j5. This sequence has to be operated five times in the same order because each job has three operations. So that the initial seed as the following format [5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1]. Each job has five operations, and each operation must run on all five machines with a certain time period.

Table 3: Processing Time

Processing Time (Sec)					
	MACHINE				
JOB	**1**	**2**	**3**	**4**	**5**
J1	64	7	74	54	80
J2	66	69	70	45	45
J3	31	68	60	98	10
J4	85	14	1	76	15
J5	44	18	90	13	91

Table 4: Machine Sequence

MACHINE SEQUENCE					
JOB	**1**	**2**	**3**	**4**	**5**
J1	m1	m2	m3	m4	m5
J2	m1	m3	m4	m5	m1
J3	m3	m4	m5	m1	m2
J4	m2	m5	m1	m4	m3
J5	m5	m1	m2	m3	m1

There are three 2s in the seed, which stands for the three operations of job j2. The first 2 corresponds to the first operation of job j2 which will be processed on machine 1, the second 2 corresponds to the second operation of job j2 which will be processed on machine 3, and the third 2 corresponds to the third operation of job j2 which will be processed on machine 2. We can see that all operations for job j2 are given the same symbol 2 and then interpreted according to their orders of occurrence in the sequence of this seed. This concept is used to find the makespan for the sequences of the problems where the generated seed (job sequence) is operated equally to the number of machines represented in the particular problem.

D. Case Example

Lawrence (LA16) n number of Jobs 10 x m number of machines 10 problems is considered. Table 5 and Table 6 shows the operation sequence and its corresponding processing time [40].

Table 5: Operation Sequence Job Shop

1	6	9	8	7	2	0	4	3	5
4	2	5	9	0	7	1	8	6	3
3	2	8	1	4	9	7	6	0	5
1	3	2	7	8	9	6	0	5	4
2	0	5	6	7	1	4	9	3	8
2	3	5	9	4	6	0	8	1	7
3	2	0	1	9	8	6	5	4	7
1	0	3	4	6	9	8	5	2	7
4	2	8	5	3	7	1	6	9	0
8	9	2	4	3	0	7	6	1	5

Table 6: Processing Time of the Jobs

21	71	16	52	26	34	53	21	55	95
55	31	98	79	12	66	42	77	77	39
34	64	62	19	92	79	43	54	83	37
87	69	87	38	24	83	41	93	77	60
98	44	25	75	43	49	96	77	17	79
35	76	28	10	61	9	95	35	7	95
16	59	46	91	43	50	52	59	28	27
45	87	41	20	54	43	14	9	39	71
33	37	66	33	26	8	28	89	42	78
69	81	94	96	27	69	45	78	74	84

Table 7: Initial Operation Sequence Job Shop

1	1	8	6	2	3	9	7	4	5
4	1	9	2	7	6	5	10	8	3
3	7	1	2	9	1	8	4	6	5
1	6	7	3	9	5	2	8	10	4
2	4	6	10	1	3	5	7	9	8
2	1	9	3	6	1	5	4	8	7
3	6	1	2	8	4	10	9	5	7
1	8	4	10	9	2	3	6	5	7
4	1	5	2	7	9	8	3	6	10
8	7	4	9	1	1	2	3	6	5

Table 8: Initial Processing Time of the Jobs

21	53	52	71	34	55	1	2	21	95
55	42	79	31	66	77	9	1	77	39
34	43	19	64	79	83	6	9	54	37
87	41	38	69	83	77	8	2	93	60
98	96	75	44	49	17	2	4	77	79
35	95	10	76	9	7	2	6	35	95
16	52	91	59	50	28	4	4	59	27
45	14	20	87	43	39	4	5	9	71
33	28	33	37	8	42	6	2	89	78
69	45	96	81	69	74	9	2	78	84

Initiations

Initial population of sequence generated randomly by increasing order and selected sequence divided into number of memeplexes are shown in Table 7 & Table 8.

Local Search

The division is done with the high -level frog (column sequence) arranged in the first memplex, the second one arranged in the second memplex, the last frog to the last memplex and repeated frog back to the next order memplex are shown in Table 9. Fitness function evaluated within the limits that the memplex are infeasible.

Table 9 Column Sequence Arrangement

8	6	9	4	10	5	1	2	3
9	2	4	1	6	5	2	3	8
1	7	5	8	4	1	10	**6**	3
1	5	7	3	**10**	8	2	**6**	4
3	7	9	8	**10**	1	5	4	6
4	3	7	9	5	8	10	6	2
1	5	1	9	7	4	2	3	8
2	5	3	1	6	8	10	7	4
6	8	4	2	9	10	7	5	1
8	6	9	4	10	5	1	3	2

Global Information Exchange

The best frog memplex values were identified each subset was compared to each other to produce best sequence way of schedule.

Iterations

For each iteration process, the frogs with the best fitness and worst fitness were identified and also the frog with the makespan schedule was identified. Finally, if the convergence criteria are not satisfied the position of the worst frog for the memplex is adjusted and new subsets of memplex will be created for the next iteration. This procedure is repeated for the desired number of iterations to reach the optimal result.

VI. NUMERICAL ILLUSTRATION

Stage 1: Initiations

The initial population of job seed sequences is generated randomly by increasing order and selected sequence divided into the number of memeplexes.

Sequence No. 1

[(2-1-8-10-6-3-9-7-4-5)(7-4-9-1-2-6-5-10-8-3)(10-3-1-9-2-7-8-4-6-5)(1-2-6-4-10-3-5-7-9-8)(6-10-9-2-3-1-5-4-8-7)(9-1-7-6-3-5-2-8-10-4)(8-3-1-6-2-4-10-9-5-7)(4-9-10-8-1-2-3-6-5-7)(7-4-5-1-2-9-8-3-6-10)(2-1-8-10-6-3-9-7-4-5)]

THC (with Tardy Job) = 167345.78 Makespan = 946

THC (with no Tardy Job)= 159875.543

Sequence No. 2

[(9-1-7-6-3-5-2-8-10-4)(7-4-9-1-2-6-5-10-8-3)(10-3-1-9-2-7-8-4-6-5)(1-2-6-4-10-3-5-7-9-8)(6-10-9-2-3-1-5-4-8-7)(2-1-8-10-6-3-9-7-4-5)(8-3-1-6-2-4-10-9-5-7)(4-9-10-8-1-2-3-6-5-7)(7-4-5-1-2-9-8-3-6-10)(2-1-8-10-6-3-9-7-4-5)]

THC (with Tardy Job) = 167316.456 Makespan = 945 THC (with no Tardy Job) = 159645.563

Sequence No. 3

[(7-4-9-1-2-6-5-10-8-3)(2-1-8-10-6-3-9-7-4-5)(10-3-1-9-2-7-8-4-6-5)(1-2-6-4-10-3-5-7-9-8)(2-1-8-10-6-3-9-7-4-5)(6-10-9-2-3-1-5-4-8-7)(9-1-7-6-3-5-2-8-10-4)(9-1-7-6-3-5-2-8-10-4)(8-3-1-6-2-4-10-9-5-7)(7-4-5-2-9-8-3-6-10)(4-9-10-8-1-2-3-6-5-7)]

THC (with Tardy Job) = 167254.663 Makespan = 943 THC (with no Tardy Job)= 159573.423

Sequence No. 4

[(1-4-9-7-2-6-5-10-8-3)(10-1-8-2-6-3-9-7-4-5)(9-3-1-10-2-7-8-4-6-5)(4-2-6-1-10-3-5-7-9-8)(8-1-2-10-6-3-9-7-4-5)(6-10-9-2-3-1-5-4-8-7)(9-1-7-6-3-5-2-8-10-4)(9-1-7-6-3-5-2-8-10-4)(8-3-1-6-2-4-10-9-5-7)(7-4-5-2-9-8-3-6-10)(4-9-10-8-1-2-3-6-5-7)]

THC (with Tardy Job) = 167232.68 Makespan = 944

THC (with no Tardy Job) = 159426.75

Sequence No. 5

q[(7-2-3-9-7-10-4-5-6-8)(8-10-9-2-7-3-6-5-4-1)(2-9-5-8-7-6-10-1-3-4)(1-3-5-4-10-6-2-7-9-8)(3-5-10-9-7-1-4-2-8-6)(3-5-10-9-7-1-4-2-8-6)(8-3-4-2-7-10-6-5-9-1)(9-2-3-7-6-10-1-8-4-5)(3-5- 4-10-6-2-7-9-8-1)(5-10-9-7-1-4-2-3-8-6)]

THC (with Tardy Job) = 168765.42 Makespan = 942

THC (with no Tardy Job) = 156456.75

Sequence No. 6

[(10-5-3-9-7-1-6-2-8-4)(7-3-4-2-8-10-5-6-9-1)(2-9-5-3-7-6-10-4-8-1)(1-3-5-6-10-4-2-7-9-8)(9-5-10-3-7-1-4-2-8-

6)(3-5-4-9-7-1-10-2-8-6)(8-3-4-2-7-10-5-6-9-1)(3-2-9-7-6-10-1-8-4-5)(3-5- 4-10-6-2-7-9-8-1)(5-10-9-7-1-4-2-3-8-6)]

THC (with Tardy Job) = 165235.42 Makespan = 941

THC (with no Tardy Job) = 156224.21

Sequence No. 7

[(4-9-10-8-1-2-3-6-5-7)(10-1-8-2-6-3-9-7-4-5)(9-3-1-10-2-7-8-4-6-5)(4-2-6-1-10-3-5-7-9-8)(8-1-2-10-6-3-9-7-4-5)(6-10-9-2-3-1-5-4-8-7)(9-1-7-6-3-5-2-8-10-4)(9-1-7-6-3-5-2-8-10-4)(8-3-1-6-2-4-10-9-5-7)(7-4-5-2-9-8-3-6-10)(8-3-4-2-7-10-5-6-9-1)]

THC (with Tardy Job) = 166522.64 Makespan = 945

THC (with no Tardy Job) = 157523.36

Sequence No. 8

[(9-10-1-7-2-4-3-6-5)(8-5-9-4-6-1-3-2-10-7)(2-8-9-3-7-5- 4-6-10-1)(1-3-5-6-10-4-2-7-9-8)(9-5-10-3-7-1-4-2-8-6)(3-5-4-9-7-1-10-2-8-6)(8-3-4-2-7-10-5-6-9-1)(3-2-9-7-6-10-1-8-4-5)(8-3-4-2-7-10-6-5-9-1)(9-2-3-7-6-10-1-8-4-5)]

THC (with Tardy Job) = 165823.35 Makespan = 941

THC (with no Tardy Job) = 154424.99

Sequence No. 9

[(9-7-4-3-1-10-5-6-8-5)(2-5-6-8-9-2-6-7-4-1)(7-5-10-1-2-6-9-8-3-4)(9-2-1-6-8-3-7-4-5-10)(5-6-4-8-10-7-3-9-2-1)(8-4-6-9-3-1-10-2-7-5)(5-1-8-6-9-2-10-7-3-4)(8-4-6-9-3-1-10-2-7-5)(3-5-4-9-7-1-10-2-8-6)]

THC (with Tardy Job) = 164667.35 Makespan = 940

THC (with no Tardy Job)= 154298.99

Sequence No. 10

[(2-4-6-1-3-5-9-10-8-7)(7-5-6-9-10-4-5-2-3-1)(4-5-7-9-10-1-2-4-3-5)(3-2-9-7-6-10-1-8-4-5)(9-1-7-6-3-5-2-8-10-4)(2-1-5-3-7-10-8-9-6)(2-8-9-3-7-4-5-1-10-6))(10-8-9-7-3-5- 4-6-10-2)(4-2-6-1-10-3-8-7-9-5)(7-1-2-10-6-3-9-8-4-5)

THC (with Tardy Job) = 165543.35 Makespan = 941

THC (with no Tardy Job)= 154875.76

Stage 2: Population Creation

For each individual population i P calculate the fitness function f (i). Based on the fitness function calculate the size of each memeplex subsets and also randomly generate the population of the job sequence. The fitness function is 5, their sequences memeplexes are (2, 5) & (6,8,10). The next step of operation sequences are grouped randomly.

Set 1	Set 2
Sequence No. 2	Sequence No. 1
Sequence No. 5	Sequence No. 7

Sequence No. 6 Sequence No. 3

Sequence No. 8 Sequence No. 5

Sequence No. 10 Sequence No. 4

Stage 3: Mutation

In the mutation operation, memeplex subsets are generated using the mutation strategy to find the population P in descending order based on their fitness. Then evaluate and Divide the population sequence P into m memeplexes with consideration of populations which is selected randomly. Two random positions are chosen and population mutation is performed in between two positions in set 2. 5^{th} and 7^{th} sequences positions are chosen randomly and inverse mutation is performed. Their next sequence orders are (1,5,7,3,4)& (2,8,5,6,10)

Table 10: Mutant Operation Sequence

<table>
<tr><td>8</td><td>7</td><td>6</td><td>9</td><td>4</td><td>10</td><td>5</td><td>1</td><td>3</td><td>2</td><td rowspan="10">Make span:
938
THC
(with Tardy
Jobs): 160248.14
THC(with Tardy
Jobs): 158722.76</td></tr>
<tr><td>2</td><td>9</td><td>7</td><td>6</td><td>2</td><td>5</td><td>3</td><td>1</td><td>4</td><td>8</td></tr>
<tr><td>9</td><td>10</td><td>7</td><td>6</td><td>10</td><td>1</td><td>4</td><td>8</td><td>5</td><td>3</td></tr>
<tr><td>9</td><td>1</td><td>5</td><td>7</td><td>3</td><td>10</td><td>8</td><td>2</td><td>6</td><td>4</td></tr>
<tr><td>2</td><td>3</td><td>7</td><td>9</td><td>8</td><td>10</td><td>1</td><td>5</td><td>4</td><td>6</td></tr>
<tr><td>1</td><td>4</td><td>3</td><td>7</td><td>9</td><td>5</td><td>8</td><td>10</td><td>6</td><td>2</td></tr>
<tr><td>6</td><td>10</td><td>5</td><td>1</td><td>9</td><td>7</td><td>4</td><td>2</td><td>3</td><td>8</td></tr>
<tr><td>9</td><td>2</td><td>5</td><td>3</td><td>1</td><td>6</td><td>8</td><td>10</td><td>7</td><td>4</td></tr>
<tr><td>3</td><td>6</td><td>8</td><td>4</td><td>2</td><td>9</td><td>10</td><td>7</td><td>5</td><td>1</td></tr>
<tr><td>7</td><td>8</td><td>6</td><td>9</td><td>4</td><td>10</td><td>5</td><td>1</td><td>3</td><td>2</td></tr>
</table>

Stage 4: Crossover

Mutation operation generators are used to generate a trial function vector. In this operation, a random population sequence is generated in between 0 to 1 and if the random number is less than the crossover constant value copy the target value otherwise the mutant operation sequence value will be changed as 0 or 1for i=1 to the number of generations are shown in Table.10.

Stage 5: Local Search

The division is done with the high- level frog (column sequence) arranged in the first memplex, the second one arranged in the second memplex, the last frog to the last memplex and repeated frog back to the next order memplex. Fitness function evaluated within the limits that the memplex are infeasible. Then Perform the Local search to improve the frog position to have the best food.

Stage 6: Global Information Exchange

The best frog memplex values were identified with each subset was compared to each other to produce best sequence way of schedule. For each memeplex, determine the best and worst frogs and improve the worst frog position by removing worst frogs in the operation sequence frame.

Stage 7: Shuffling

The trial sequence obtained by the crossover operation generation is compared with the target sequence to determine the jobs and machine schedule that participates in the next generation and the fittest is passed on to the next generation. If the objective value f (i) is lower than required processing value Pi.), then random value replaces the best compared value, Otherwise, consider the best fitness. Finally, Shuffle each improved memeplexes and Combine the evolve memeplexes and also Sort the population P in descending order of their best fitness value.

Stage 7: Iterations

For each iteration, the frogs with the best fitness and worst fitness were identified and also the frog with the makespan schedule was identified.

Stage 8: Control Parameters

Finally, if the convergence criteria are not satisfied the position of the worst frog for the memplex is adjusted and new subsets of memplex will be created for the next iteration. This procedure is repeated for the desired number of iterations to reach the optimal result.

A. Final Result obtained using SFLA Algorithm

The best solutions found in 100 iterations of the local search process and Global information exchange for minimizing total holding cost with tardy jobs and without consideration of tardy jobs are listed in Table 11.

Case (i) Consideration of all production tasks with the Same Function

Case (ii) Consideration of all production tasks with the individual functions results are shown in Table 12.

Table 11: Results Obtained after 100 Iterations

SFLA	Local Search			Glopal Information Exchange		
Iteration's	Makespan	Total Holding Cost with Tardy Jobs	Total Holding Cost with no Tardy Jobs	Makespan	Total Holding Cost with Tardy Jobs	Total Holding Cost with no Tardy Jobs
First Iteration	944	162557.0299	159798.6	944	161357.091	151798.6
Second Iteration	942	161890.5434	158548.6	945	161093.534	152473.6
Third Iteration	940	161224.0569	156595.6	943	160829.977	151346.7
Last Iteration	940	159557.5704	154665.6	939	158192.673	150268.6

Table 12: Individual Function Considerations

Indivual Function Consederations								
Description	Makespan	Mean Tardiness Cost	Mean Earliness Cost	Finished Product Holding Cost	Inprocess Holding Cost	Only THC	THC with Tardi	THC with all
Notations		∑ti	∑Ei	∑Pi	∑Ii	∑Hij	∑hiti	c(s)
Sequence No. 1	947	121.89	462.534	64550	97790.06	162340.06	162461.95	161999.416
Sequence No. 2	946	131.3	642.643	63299	98772.4	162071.4	162202.7	161560.057
Sequence No. 3	945	131.4	432.335	63433	98676.13	162109.13	162240.53	161808.195
Sequence No. 4	944	141.7	462.305	62573.6	99298.93	161872.53	162014.23	161551.925
Sequence No. 5	943	141.95	447.2055	61975.2	99741.96	161717.16	161859.11	161411.9045
Sequence No. 6	942	162.2	426.106	61736.7	100185	161921.7	162083.9	161657.794
Sequence No. 7	941	162.7	405.0065	60738.1	100628.038	161366.138	161528.838	161123.8315
Sequence No. 8	940	169.7	383.907	60169.3	101071.07	161240.37	161410.07	161026.163
Sequence No. 9	938	176.71	362.8075	59671.4	101514.1083	161185.5083	161362.2183	160999.4108
Sequence No. 10	936	183.6	341.708	59032.6	101957.14	160989.74	161173.34	160831.632

B. Practical Applications of Proposed SFL Algorithm for Minimizing THC with and Without Tardy Jobs

Table 13: Customer Order

Date of Order	Product Required	Needed Quantity	Due Date (week)
25/03/2012	Finned Tubes	1000	1
		1500	3
		1400	4
		2000	5
		1400	6
		3000	8

The proposed SFL algorithm can be successfully implemented in industries handling a wide variety of products in small volumes and the industries working with general purpose machines which can handle different operations. The job processing and waiting times can be conveniently cut down, machine loads can be balanced and also the user has a choice of choosing a solution from the set of alternative solutions as per his desired objective criteria.

Customer Order

Table.13 and Figure.1 shows the customer order for manufacturing finned tubes. It gives details of needed quantity in terms of the week.

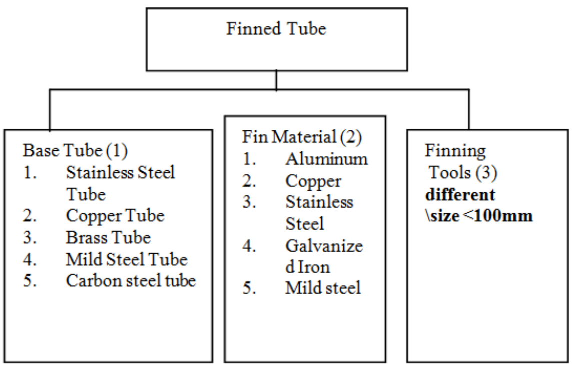

Figure 1: Product Structure of Finned Tubes

Inventory Record Details

The Inventory record details for base Tube (1) is shown in Table 14 and for Fin, material is shown in Table 15. The final results obtained after 100 iterations for the case study problem is listed in Table 16. Minimizing the total holding cost, Meeting due dates, Minimizing tardiness, Earliness are the most important goal of scheduling to avoid the delay penalties including customer's bad impression, lost future sales, etc.. Many research papers have focused on due-date oriented functions, whereas the main aim of optimizing the makespan is to minimizing costs and maximizing the output. In Many research articles and studies, Heuristics Algorithms are competent and proves to be a good problem-solving technique for job shop scheduling.

Table 14: Inventory Record Details for Base Tube (1)

Job/Part No	Part Description (Fin Strip)	Lead Time (week)	Stock on Hand(size)
j_1	Stainless Steel	3 weeks	Min 6mm to Max 15mm in 150 kg
j_2	Copper Tube	4 weeks	Min 6mm to Max 15mm in 200 kg
j_3	Brass Tube	5 weeks	Min 6mm to Max 15mm in 150 kg
j_4	Mild Steel Tube, Galvanized Iron	2 Weeks	Min 6mm to Max 15mm in 100 kg
j_5	Aluminum, Copper, Mild Steel	3 Weeks	Min 6mm to Max 15mm in 350 kg

Table 15: Inventory Record Details for Fin Material (2)

Job/ Part No	Part Description	Lead Time (week)	Stock on hand
j_1	Stainless Steel Tube	3 weeks	3m in 200 nos 2.5m in 50 nos
j_2	Copper Tube Aluminium	4 weeks	3 m in 100 nos 6 m in 150 nos
j_3	Copper Tube Aluminium	5 weeks	3 m in 50 nos 6 m in 100 nos
j_4	Aluminium	2 Weeks	6 m in 50 nos
j_5	Carbon Steel Tube	3 Weeks	6m in 350nos

C. Benchmark Problem Solutions

Five instances of size (n X m = 10 X 10) denoted as (LA16-LA20) from Lawrence [56] with different 't' parameter for to control the due dates. Setting t = 2.5, 3.5 where the due-date constraint is loose and t = 1.8, 1.9 where the due-date constraint is strict.

Benchmark problem which contains Ten jobs and ten machines for instances LA 16- LA 20 taken from Lawrence was tested with the proposed SFL algorithm and the results are compared with Sheep Frog Leaping Algorithm, Artificial immune system and Heuristic Shifting Bottleneck (HSB) procedure reported in the literature (Reference). These instances were tested for minimum makespan problem, Total holding cost problem and due date consideration problem. The best objective function value solution, computing time is investigated for single objective function and separate objective functions. The results are shown in Table 17 and Table 18. Furthermore, in order to investigate the performance of the proposed heuristic approaches for more problems, the LA36-LA40 of 15 jobs and 15 machines benchmark problems are also considered. The literature heuristics procedures are needs the enormous computation time to solve the 15 jobs and 15 machines problems. The proposed SFL algorithm has achieved good solutions for the 15 jobs and 15 machines problems with lesser computing time.

Table 16: Final Results Obtained for the Case Study Problem

Indivual Function Consederations								
Description	Makespan	Mean Tardiness Cost	Mean Earliness Cost	Finished Product Holding Cost	Inprocess Holding Cost	Only THC	THC with Tardi	THC with all
Notations		∑ti	∑Ei	∑Pi	∑Ii	∑Hij	∑hiti	c(s)
Sequence No. 2	188.12	11.09	2.106	59356.6	61842.6816	121199.282	121210.372	121208.2656
Sequence No. 3	188.06	10.99	2.0065	58820.5	61246.6433	120067.173	120078.163	120076.1568
Sequence No. 4	184.6	10.59	1.907	58704.46	61208.605	119913.065	119923.655	119921.748
Sequence No. 5	183.785	10.265	1.8075	57268.4	60470.5667	117738.957	117749.222	117747.4142
Sequence No. 6	182.252	10.18	1.708	56652.32	60232.5284	116884.848	116895.028	116893.3204
Sequence No. 7	180.719	10.095	1.6085	55376.3	59294.4901	114670.74	114680.835	114679.2266
Sequence No. 8	179.186	10.01	1.509	55070.18	59516.4518	114586.632	114596.642	114595.1328
Sequence No. 9	177.653	9.925	1.4095	54224.1	59118.4135	113342.524	113352.449	113351.039
Sequence No. 1	176.12	9.84	1.31	53648.04	58580.3752	112228.415	112238.255	112236.9452

Table 17: Ten Jobs and Ten Machines for Instances LA 16- LA 20

Heuristics Algorithm	Problem	t = 1.8		t = 1.9		t = 2.5		t = 3.5	
		Best Value	CPU time in sec	Best Value	CPU time in sec	Best Value	CPU time in sec	Best Value	CPU time in sec
SFHM Algorithm	**LA 16**	155458	14	162765	11.5	181715	19.6	219945	11.5
	LA 17	140071	2.8	140902	10.8	164089	4.5	201687	19.8
	LA 18	141624	21.5	139986	18.6	159937	17.2	189851	17.3
	LA 19	112760	30.8	111971	21	120963	28.6	133792	23.7
	LA 20	113499	9.6	170635	14.5	195046	26.5	244373	21.9
AIS Algorithm	**LA 16**	160112	25	164354	18	184679	29.5	220567	15.7
	LA 17	144894	3	146106	19.4	166005	6	202075	32.3
	LA 18	143004	29.7	143348	30.9	160568	21.4	190933	21.4
	LA 19	114057	38.5	112977	34.5	122901	44.8	134027	44
	LA 20	115973	12.1	171012	20.7	197634	40.2	245416	38.6
HSB Algorithm	**LA 16**	160374	28.2	165402	25.2	186995	32.7	221604	19.1
	LA 17	146206	3.2	147518	25.2	167567	6.3	202933	38.2
	LA 18	143833	33.4	145730	35.1	162154	22.9	191316	22.2
	LA 19	114242	43.7	115251	49	123436	56.7	135171	57.8
	LA 20	116339	13.8	171343	25.5	199713	42.6	247715	49.3
SFL Algorithm	***LA 16***	***155458***	***14***	***161358***	***10.8***	***173433***	***18.4***	***214556***	***11.1***
	LA 17	***139871***	***3.9***	***140589***	***10.5***	***163662***	***4.3***	***201124***	***18.6***
	LA 18	***124284***	***21.1***	***135417***	***17.8***	***158252***	***16.8***	***187422***	***16.3***
	LA 19	***111697***	***29.3***	***111738***	***20.9***	***120534***	***27.5***	***132597***	***226***
	LA 20	***112339***	***9.5***	***168905***	***13.2***	***194872***	***25.2***	***241212***	***20.3***

Table 18: Final Result comparisons of SFLA with other Heuristics

Results Comparision of SFLA with SFHM, AIS and HSB													
Sl.No	Instance	Problem Size (Ji x Mi)	***SFLA***	SFHM	***IP***	% of Improvement over SFHM	AIS	***IP***	% of Improvement over AIS	HSB	***IP***	% of Improvement over HSB	Overall % of Improvement(IP)
1	LA 16	10x10	***939***	945	***6***	5.96	946	**7**	6.95	948	**9**	8.91	7.27
2	LA 17	10x10	***783***	784	***1***	1.00	791	**8**	7.92	789	**6**	5.95	4.96
3	LA 18	10x10	***846***	848	***2***	2.00	847	**1**	1.00	849	**3**	2.99	1.99
4	LA 19	10x10	***841***	842	***1***	1.00	851	**10**	9.88	853	**12**	11.83	7.57
5	LA 20	10x10	***898***	902	***4***	3.98	899	**1**	1.00	901	**3**	2.99	2.66
6	LA 36	15x15	***1266***	1268	***2***	2.00	1268	**2**	2.00	1273	**7**	6.96	3.65
7	LA 37	15x15	***1389***	1401	***12***	11.90	1405	**16**	15.82	1397	**8**	7.95	11.89
8	LA 38	15x15	***1196***	1196	***0***	0.00	1196	**0**	0.00	1196	**0**	0.00	0.00
9	LA 39	15x15	***1233***	1233	***0***	0.00	1234	**1**	1.00	1233	**0**	0.00	0.33
10	LA 40	15x15	***1223***	1224	***1***	1.00	1227	**4**	3.99	1238	**15**	14.82	6.60

VII. CONCLUSIONS

To avoid customer's bad impression and To improve the customer's Satisfaction by delivering the jobs within the due date is a very important criterion in the manufacturing system. In order to avoid delay penalties including customer's bad impression, cost of lost future sales and rush shipping cost, due date constraints are considered. The objective considered in this paper is minimizing total holding cost which means a sum of product inventory cost and in process inventory cost with consideration of tardy jobs and without consideration of tardy jobs. And also a heuristic Shuffled frog leaping algorithm is proposed for the job-shop scheduling problem to minimize total holding cost and make the pan. Strict due date parameter and lose due date parameter is used for analyzing the total holding cost. The proposed heuristics are used for testing Lawrence 10 x 10 and 15 x 15 benchmark problems. Results show that SFL algorithm produces good quality results compared with other Heuristics approach procedures.

REFERENCES

1. *Ahmed S. Ghoniem, "Static and dynamic job-shop scheduling using rolling horizon approaches and the Shifting Bottleneck Procedure "Thesis, Master of Science In Industrial and Systems Engineering (2002)*
2. *Akers, S. B., (1956), "A Graphical Approach to Production Scheduling Problems", Operations Research, vol 4, pp. 244-245.*
3. *Hefetz, N., and Adiri, I., (1982), "An Efficient Optimal Algorithm for the Two- Machines Unit-Time Job-Shop Schedule-Length Problem", Mathematics of Operations Research, vol 7, pp. 354-360.*
4. *Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., and Shmoys, D. B., (1997), "Short Shop Schedules", Operations Research, 45(2), pp. 288-294*
5. *Van Laarhoven P.J.M, Aarts E.H.L, Lenstra J.K, Job shop scheduling by simulated annealing. Operational Research, 40:113–125, 1992.*
6. *Lenstra, J. K., Rinnooy Kan, A. H. G., and Brucker, P., (1977), "Complexity of Machine Scheduling Problems", Annals of Discrete Mathematics, vol 7, pp. 343-362.*
7. *N. Kamatchi et al.,, A Hybrid Algorithm using Firefly and Cuckoo Search Algorithm for Flexible Open Shop Scheduling Problem, International Journal of Mechanical and Production Engineering Research and Development (IJMPERD), Volume 7, Issue 2, March- Apri;2017, pp. 155-166*
8. *Lenstra, J.K., and Rinnooy Kan, A. H. G., (1979), "Computational Complexity of Discrete Optimization Problems", Annals of Discrete Mathematics, vol 4, pp. 121-140.*

9. *Mattfeld, D.C., Bierwirth, C., and Kopfer, H., (1998), "A Search Space Analysis of the Job Shop Scheduling Problem", to appear in Annals of Operations Research.*
10. *Carlier J, Pinson E, An algorithm for solving the job-shop problem. Manage Sci, 35: 164–176, 1989.*
11. *Shmoys, D. B., Stein, C., and Wein, J., (1994), "Improved Approximation Algorithms for Job Shop Scheduling Problems", SIAM Journal of Computer science, June, 23(3), pp. 617-632.*
12. *French, S., (1982), "Sequencing and Scheduling - An Introduction to the Mathematics of the Job- Shop", Ellis Horwood, John-Wiley & Sons, New York.*
13. *Erschler, JF, Roubellat, JP, Vernhes, (1976), "Finding some essential of the feasible solutions for a scheduling problem", Operations Research, 24, pp. 774- 783.*
14. *Brucker, J. R., and McMahon, G. B., (1985), "Scheduling the General Job-Shop", Management Science, May, 31(5), pp. 594-598.*
15. *Brooks, G. H., and White, C. R., (1965), "An Algorithm for Finding Optimal or Near Optimal Solutions to the Production Scheduling Problem", Journal Of Industrial Engineering, January, 16(1), pp. 34-40.*
16. *Lageweg, B. J., Lenstra, K. and Rinnooy Kan, A. H. G. (1977) Job-Shop Scheduling by Implicit Enumeration, Management Science, December, 24(4), 441-450.*
17. *Jackson, J.R., (1957), "Simulation Research on Job-Shop Production", Naval Research Logistics Quarterly, vol 4, pp. 287-295.*
18. *Panwalkar, S.S., and Iskander, W., (1977), "A Survey of Scheduling Rules", Operations Research, Jan-Feb, 25(1), pp. 45-61.*
19. *Haupt, R. (1989), A Survey of Priority Rule Based Scheduling, OR Spektrum, vol 11, pp. 3-16.*
20. *Chang, Y.L, Sueyoshi, T., and Sullivan, R.S., (1996), "Ranking Dispatching Rules by Data Envelopment Analysis in a Job-Shop Environment", IIE Transactions, 28(8), pp. 631-642.*
21. *Lawrence, S., (1984), "Supplement to Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques", Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, USA, October.*
22. *Manne, A. S., (1960), "On the Job-Shop Scheduling Problem", Operations Research, vol 8, pp. 219-223.*
23. *Blazewicz, J., Dror, M., and Weglarz, J., (1991), "Mathematical Programming Formulations for Machine Scheduling: A Survey", European Journal of Operational Research, 51(3), pp. 283-300*
24. *M.Chandrasekaran, P.Asokan, S.Kumanan, " Multi objective optimization for job shop scheduling problems using Sheep Flocks Heredity Modal Algorithm", Int J Mfg Sci and Tech, Vol 9, 2:47-54, 2007.*
25. *M.Eusuff, K.Lansey, F.Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization". Eng Optima 38:129–154, 2006.*
26. *Mohammadreza Farahani, Saber Bayat Movahhed, Seyyed Farid Ghaderi1, A hybrid meta- heuristic optimization algorithm based on SFLA. 2nd International Conference on Engineering Optimization, September 6 - 9, 2010, Lisbon, Portugal*
27. *Quan-Ke Pan & Ling Wang & Liang Gao & Junqing Li, An effective shuffled frog-leaping algorithm for lot-streaming flow shop scheduling problem.Int J Adv Manufacturing Technology, DOI 10.1007/s00170-010-2775-3*

28. *Erschler JF, Roubellat JP, Vernhes, Finding some essential characteristics of the feasible solutions for a scheduling problem. Oper Res, 24:774-783, 1976.*

29. *Eusuff M, Lansey K, Pasha F (2006) Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. Eng Optim 38:129–154*

30. *Mohammadreza Farahani, Saber Bayat Movahhed, Seyyed Farid Ghaderi1, A hybrid meta- heuristic optimization algorithm based on SFLA. 2nd International Conference on Engineering Optimization September 6 - 9, 2010, Lisbon, Portugal*

31. *M. Eusuff, K. E. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, Journal of Water Resource Planning and Management, vol. 129, issue 3, 210-225,2003*

32. *Balas, E., Lenstra, J.K., and Vazacopoulos, A., (1995), "The One-Machine Problem with Delayed Precedence Constraints and its use in Job Shop Scheduling", Management Science, Jan, 41(1), pp. 94-109.*

33. *Emad Elbeltagi, Tarek Hegazyz and Donald Griersonmad (2007), "A modified shuffled frog- leaping optimization algorithm applications to project management" Structure and Infrastructure Engineering, Vol. 3, No. 1, March 2007, 53 – 60*

34. *Quan-Ke Pan & Ling Wang & Liang Gao & Junqing Li, An effective shuffled frog-leaping algorithm for lot-streaming flow shop scheduling problem.Int J Adv Manufacturing Technology, DOI 10.1007/s00170-010-2775-3*

35. *M. Alinia Ahandani, N. Pourqorban Shirjoposht & R. Banimahd, Job-Shop Scheduling Using Hybrid Shuffled Frog Leaping, online*

36. *Emad Elbeltagia,, Tarek Hegazy and Donald Grierson, Comparison among five evolutionary- based optimization algorithms, Advanced Engineering Infomatics 19 (2005),43-53*

37. *Ziyang Zhen, Daobo Wang and YaunYaun (2009) " Improved Shuffled Frog Leaping Algorithm for Cointinuous optimization problem", Proceeding CEC,09 of the Eleventh Conference on congress on Evolutionary Computation Pages 2992 – 2995, IEEE press Piscataway NJ, USA. ISBN : 978-1-4244-2958-5*

38. *Chen Fang and Ling Wang, An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem, Journal Computers and Operations Researh, Volume 39 Issue 5, May, 2012*

39. *Kerem Bu¨lbu¨l, Philip Kaminsky and Candace Yano(2004), "Flow Shop Scheduling with Earliness, Tardiness, and Intermediate Inventory Holding Costs" Naval Research Logistics, Vol. 51 (2004), DOI 10.1002/nav.20000*

40. *Hiroshi Ohta, Toshihiro Nakatani, A heuristic job-shop scheduling algorithm to minimize the total holding cost of completed and in-process products subject to no tardy jobs. Int J Prod Eco, 101:19–29 (2006)*

41. *S.Lawrence, Supplement to Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Tech., Carnegie Mellon University, GSIA (1984)*