Flexible Job Shop Scheduling using Hybrid Swarm Intelligence

S. Kavitha, P. Venkumar

Abstract: In the present environment investigation, one of the essential tasks to be solved is scheduling. The most significant issue in the Job Shop scheduling process is the flexibility which is occurred during the manufacturing process. This paper presents the hybridization of swarm intelligence’s that is Chicken Swarm Optimization (CSO) and Discrete Fish Swarm Optimization (DFS0) to minimize the makespan, overall workload and utmost workload of the machine. The purpose of individual operators is employed to upgrade the fish position and provoke new fishes that are processing times. The purpose of this technique is to speed up the minimum convergence and trapped in the local optimum. The proposed hybrid algorithm results are compared with conventional and existing optimization approaches for a multi-objective flexible JSP process.

Keywords: Flexible Job shop Scheduling, Hybrid algorithm, Multi objectives, Swarm intelligence and processing Times.

I. INTRODUCTION

Flexible Job Shop Scheduling Problem (FJSSP) is an accumulation of the standard JSP that established a task to be processed every machine from the available machine set. It populates most of the challenges and complexities [1] of its predecessor JSP and is more obscured than JSP due to the expansion needs to decide the task of activities to machines [2]. To tackle the problem, more consideration was applying the intelligent algorithm in the late decades. The behavior of swarm intelligence analyzed by a few researchers and depicts some strategies for separating job shop scheduling [3]. In the previous two decades, nature - motivated algorithms keep improvements in the problems of solving optimization [4]. There are extended to comprehend an extensive variety of utilization, for example, scheduling optimization problems [5]. The probabilistic search techniques observed by most analyst and they proposed that it is an attiring another option to find solution this constrained optimization problem such as Genetic based algorithms [6], Simulated annealing algorithm, Bat algorithm [7] particle swarm optimization [8], Ants colony algorithm [9], Artificial immune algorithm, etc. An altered coding scheme was introduced to minimize the makespan measure of FJSSP [10]. In the initial phase, a novel machine task methodology was furnished and while the current solution had not been enhanced at the time an enhancement procedure was performed [11]. The main aim of this study is the scheduling steadiness in the FJSSP in that machine breakdowns are normal. The simulations of a machine breakdown can provide results in a scheduling solution with higher security, more robustness and nearest to the real world and it has not been researched in many investigations [12]. An optimal schedule neglected the assignments of tasks machines and completion time in JSSP. In this research, a hybrid algorithm is optimized for an FJSSP because it apposing multi goals like maximum workload, total workload and Markesan time JSSP is tackled by the behavior of hybrid optimization as CSO and DFSO are connected.

II. LITERATURE REVIEW

<table>
<thead>
<tr>
<th>Literature Features</th>
<th>Technique</th>
<th>Objectives</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chao Peng et al. 2018 [13]</td>
<td>GA</td>
<td>Make span Time</td>
<td>JSSP in realistic production exercises dependably needs to think about the requirements of various manufacturing assets. Both machines and workers are considered during the process of job shop scheduling.</td>
</tr>
<tr>
<td>TIANHUA JIANG et al. 2018 [14]</td>
<td>GWO</td>
<td>Compl etion Time</td>
<td>They developed a GWO algorithm with an objective of minimizing the maximum completion time.</td>
</tr>
<tr>
<td>Liang Xu et al. 2017 [15]</td>
<td>Hybrid cloud particle swarm optimizatio n (HCPSO)</td>
<td>Make span, Total Work Load and Maximum Workload</td>
<td>It is gone for reducing the completion time of jobs, total workload, and maximum workload. The weight has stable propensity and arbitraryness properties in view of the cloud model, which enhances the convergence speed, as well as maintains the variety of the populace.</td>
</tr>
<tr>
<td>Song Huang et al. 2018 [16]</td>
<td>Particle Swarm Optimizati on (PSO)</td>
<td>Make span, Total Work Load and Maximum Workload</td>
<td>The PSO is used to solve multi-objective optimization problems.</td>
</tr>
</tbody>
</table>

Revised Manuscript Received on December 15, 2019.

* Correspondence Author

S. Kavitha, Department of Mechanical Engineering, School of Automotive and Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnankovil, Tamilnadu, India, kavitha@klu.ac.in

P. Venkumar, Department of Mechanical Engineering, School of Automotive and Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnankovil, Tamilnadu, India, p.venkumar@klu.ac.in

Retrieved Number: A10531291S419/2019©BEIESP
DOI:10.35940/ijeat.A1053.1291S419
Flexible Job Shop Scheduling using Hybrid Swarm Intelligence

Kamatchi et al. 2017 [17] Hybrid optimization (Firefly and Cuckoo Search) Make span The tensile, open shop scheduling is known to be NP-hard. Cuckoo Algorithm (CA) is one of the generally utilized strategies for constrained optimization. What’s more, it gave the best outcomes in contrast with different algorithms. A drawback of CA is that they effortlessly become trapped in the neighborhood minima.

Jun-Qing Li et al. 2013 [18] Discrete artificial bee colony algorithm (DABC) Make span, Total Work Load and Maximum Workload The DABC algorithm is used to solve the scheduling problem.

Thi-Kien Dao et al. 2015 [19] Bat algorithm Make span In the analysis, 43 scheduling dataset problems were tested using the proposed bat algorithm.

In FJSSP and JSSP the remaining manuscript is structured as takes after section. Section 2 presented our approach and section 3 analyzed the implementation and comparison results. Finally, the proposed work conclusion part is given in section 4.

III. HYBRID ALGORITHM FOR FJSSP: A METHODOLOGY

The knowledge of different swarm optimization model is considered for the FJSSP process. In FJSSP, every job is executed on the machines with machining time in a specific order. In this model, each job is processed only one machine at a time. Ordinarily, for the scheduling purpose, there are M machines and N jobs utilized. The considered jobs are processed with various routes or sequences. Hence, the difficulty of scheduling jobs is based on the number of machines, jobs, and sequences of jobs. Our proposed methodology considers Multi-Objective functions in the number of jobs and machines (J*M) process, for analyzing purpose AFSo with CSO used. From this implementation and mathematical procedure to solve our objectives in FJSSP, the detailed discussion of our methodology is described in beneath sections.

A. Problem Definition

JSP is one of the combinatorial optimization problems. The sequence of operations performed in existing engines determined the scheduling problem with a minimized processing time that needs for the process completion [20]. The number of machines symbolized as ‘m’ and the different number of jobs symbolized as ‘i’, and they are produced to be scheduled. At a particular given time, each job contains a number of the process which should be executed on a particular machine. In Table 1 furnished the sample processing time.

<table>
<thead>
<tr>
<th>Jobs/Machine</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
</tr>
</thead>
</table>

Table-1: Processing Time of Benchmark problems

J1 O1 1 O1 2 O1 3 O1 4 O4 5 1 6
J2 O2 1 O2 2 O2 3 O2 4 O4 2 3
J3 O3 1 O3 2 O3 3 O3 4 O4 1 3 6
J4 O3 4 O3 10 O3 2 6 4 5 4 7

Also, the sample processing time appears in table 1. At the point, the operation will be handled in a particular machine, it is feasible for the operations to queue up ahead of time so that the machine is being utilized by another operation.

B. Function for FJSSP Objective

Proposed FJSSP process considered three objectives such as Make span time (F1), total workload (F2), and Machine Work Load (F3). For the investigation, a set of machines and jobs, it’s denoted by \( m = \{ m_1, m_2, m_3, \ldots, m_n \} \) and \( j = \{ j_1, j_2, j_3, \ldots, j_n \} \) respectively. In FJSSP, each job \( j \) consists of a sequence of \( n_j \) operations that is \( \{ O_{j1}, O_{j2}, O_{j3}, \ldots, O_{nj} \} \) each operation has to be executed to finish a job. All objective functions constrained as a minimum its criteria followed as:

\[
\text{F1: Make span Time } \Rightarrow \text{Min} F1 = \text{Max} \{ f_{jm} \} \quad (1)
\]

\[
\text{F2: Total Workload } \Rightarrow F2 = \sum_{i=1}^{m} \sum_{j=1}^{n_i} T_{ink} \cdot p_{jk} \quad (2)
\]

\[
\text{F3: Maximum Work load } \Rightarrow F3 = \text{Max} \sum_{i=1}^{m} \sum_{j=1}^{n_i} T_{ink} \cdot p_{jk} \quad (3)
\]

The objective function \( F_{obj}(\cdot) \) i.e. fitness function of the proposed model can be written as

\[
F_{obj} = \text{MIN} \{ F1, F2, F3 \} \quad (4)
\]

Statement and Description of Our proposed FJSSP as

For easy analysis, in the assignment task, sub-problem assigns each operation to an appropriate machine. One of the sequencing sub-problems calculates an order of operations on each machine.

Furthermore, the embraced theories in this problem are:
If all the machines are ‘m’, it accessible at time zero, then all jobs are N, it also organized for processing at time zero.

There are no priority limitations in the operations of various jobs.

The operation processing time of each machine is characterized ahead of time.

During the processing time, the interruption of each machine is negligible.

Parameter Notation

\( m \) or \( M \) Number of machines
\( n \) or \( N \) Number of Jobs
\( k \) \( n \) Number of the machine in \( k \) th the job
\( O_{j} \) Operation of n machine with n jobs
\( P_{ink} \) Processing Time of each job with each machine
\( T_{fsm} \) Job completion Time for each
C. Hybridization of Swarm Optimizations

The presented swarm intelligence optimization is one of the Metaheuristic approaches which works on the principle of swarm behavior of the population. There are cooperation and competition between initialized individuals, which introduced an algorithm, attains the optimal solution among the constrained problem. The presented swarm intelligence optimization approach is an advanced technology which mainly focused by many scholars. And also the number of researchers has proposed many swarm intelligence algorithms like CSO and DFSO. The Hybridized form of CSO and DFSO is shown in figure 3 and is utilized in the proposed work to solve our objective function that is the minimum of J*M performance.

D. Chicken swarm (CS) Optimization

In the CS and the behavior of the chicken swarm, the mimics of CS optimization are in the hierarchical order derived from the perception of the winged animals foraging behavior and it is presented in figure 1.

Rooster: The chickens with the best fitness acted in this group, so it’s called as head of the CS model.

Chicks: Chickens with the worst fitness solution act as this group.

Hens: Remaining all solution, expect above groups are called hens or mother.

![Image 1. CS behavior](image)

Fig. 1. CS behavior

Generally, the relationship between these groups as mother-child with family model randomly established. The food already found by others is stolen randomly by the chickens [20], the chicken with worst fitness value is a search around their mother (a hen) for food, and the dominant individuals have benefit in the struggle for food.

Initial solution generation of j*m model shown in the below equation (5):

\[
\begin{bmatrix}
m_{11}^k & m_{12}^k & \cdots & m_{1n}^k \\
m_{21}^k & m_{22}^k & \cdots & m_{2n}^k \\
\vdots & \vdots & \ddots & \vdots \\
m_{m1}^k & m_{m2}^k & \cdots & m_{mn}^k
\end{bmatrix}
\]

Where \( m_{ij}^k \) represents the priority of operation \( O_{ij} \) is the operation of a job \( j \) that is an important process on the machine \( i \).

E. Position and Movement updating process

New solution updating process, consider the chickens as the processing time of each job and machine \( j \) (1 ≤ \( i \) ≤ \( n \)) and \( m_{kj} \) (1 ≤ \( k \) ≤ \( m \)). Considering the initial fitness evaluation, group the solution that is the Roosters \( C_R \) and worst fitness as \( C_c \) roosters with perfect fitting values have priority for food access than the ones with worse fitting values. Position updated formulated as:

F. Exploitation Phase

\[
M_{ij}^{t+1} = M_{ij}^t + (1 + \text{Rand}(0, \sigma^2))
\]

\[
\sigma^2 = \left\{ \begin{array}{ll}
1 & \text{if } P_i \leq R_f \\
\exp\left[\left(R_f - P_i \right) / \left(R_f \right) \right] & \text{otherwise}
\end{array} \right.
\]

Above updating process \( \text{Rand} (0, \sigma^2) \) is a Gaussian distribution with 0 values and \( \sigma \) as standard deviation and \( \epsilon \) as constant. Remaining group i.e. hens (Except the best and worst fitness solution) can follow their roosters for searching food and then randomly stole the food found by others. For solving the above equation (8), (9), formulated below.

G. Global Optimum Solution

\[
M_{ij}^{t+1} = M_{ij}^t + \text{Rand}^*(M_{ij}^t, L^*(M_{ij}^t)) + \text{Rand}^*(M_{ij}^t, -M_{ij}^t)
\]

(8)

\[
R_1 = \exp\left(R_f - R_{ij}) (\text{abs}(R_f) + \epsilon) \right) \quad \text{and} \quad R_2 = \exp\left(R_{ij} - R_f \right)
\]

(9)

From the above equation, the notation \( \text{Rand} \) is indicated as a uniform random number also an index of the rooster, which is the \( i \)th hen’s group-mate, while \( R_f \) ∈ \([1..N]\) is an index of the chicken (rooster or hen), which is casually selected from the swarm \( R_1 \neq R_2 \). With respect to the chicks, they always follow their mother to search for food. At final; model of minimizing \( F1, F2 \) and \( F3 \) showed in the below equation.

H. Final Local Minimum

\[
M_{ij}^{t+1} = M_{ij}^t + L^*(M_{ij}^t - M_{ij}^t)
\]

(10)

From the above equation, the notation \( M_{ij}^t \) stands for the position of the \( i \)th chick’s mother \( L \) ∈ \((0,2)\) is a parameter, which deliberates that the chick would follow its hen hunting their food. So the solution is obtained based on the encoding scheme.

I. Discrete Fish Swarm Optimization (DFSO)

Generally, a fish can modify its situation constantly with due respect to the environmental condition and the own state position. The movement of individual fish appeared randomly, yet as a sealum, it is a greatly coordinated progress towards the objective and this DFSO flowchart appeared in fig 2. Fishes stay close to the swarm in order to guard or shield fishes from the predators which means it maintains since the distance from neighboring fishes to evade collisions and endlessly search for food [21].

J. Discrete in FSO (DFSO)

In the new discrete FSO algorithm for FJSSP, a solution refers to all the cities of tour consisting which are arrayed differently and the fitness value of the solution refers to the length of the tour. The procedure of this DFSO described in the below section and this innovative model consider four important phases such as

Discrete Solution Initialization
- Fish Prey behavior
- Fish Swarm Behavior
- Fish Follow Behavior
Flexible Job Shop Scheduling using Hybrid Swarm Intelligence

(i) Solution Initialization
FJSSP processing time initialized by the size of benchmark problems and integer values based, it’s represented by equation (11).

(ii) Fish Prey Behavior
A group of fish fined food in the process of movement, the other neighbors and deploys to follow and reach the point of these possible. The fish condition analyzed in the ideal state $M_{max}$ from visual neighbors, the quantity of partner of $M_{max}$ if $(Di,j < \text{visual})$ displays that close separation has more nourishment and not very crowded additionally move to the front of $M_{max}$ position; the utilization condition conducted for performing researching generally. The sustenance fixation in this position of fish is expressed as, target capacity esteem. The separation among the artificial fish is $d_{i,j} = ||M_i - M_j||$, here $i$ and $j$ is random fish. Where creates random numbers between 0 and 1 and Step implies the greatest stride size of manufactured fish. Visual is the separate aircraft, the counterfeit fish happens just in the inward range of the hover to the length of the field of vision distinctive acts shown in equation (12) (13).

(iii) Fish Swarm Behavior
Naturally, the assembling in groups leads to guarantee the existence of the colony and to avoid dangers. Likewise, in the fish swarm behavior fishes are assembled in groups in the moving process. Let $M_i$ be the AF current state, $M_c$ be the center position and $N_c$ be the number of its companions in the current neighborhood fish is $M_i(Di,j<\text{Visual})a$ number of counterfeit fish. Based on the highest and lowest fitness functions (F1, F2 and F3) the swarm behavior updated shown in equation (14).

(iv) Fish Follow Behavior
In the process of a group of fish find food, the other neighbors are a move to follow and reach the point of these possible. The condition of fish is investigated its ideal state $M_{max}$ from Visual neighbors, the quantity of partner of $M_{max}$ if $(Di,j<\text{Visual})$ shows that close separation has more nourishment and it’s not crowded additionally move to the front of $M_{max}$ position; generally, perform searching conduct by utilizing condition (13).

(v) Move and Leave Process
Fish swim arbitrarily in water; truth be told; they are looking for food or partners in bigger ranges. The leave procedure as stop someplace in water, each AF’s behavior result will be the same, the distinction between objective values (food concentration, FC) end up litter within a few cycles, it may fall into local extreme change the parameters haphazardly to the [21] still states for leaping outflow state. These anomalous operations are said in condition (16). Swarm makes fish bound in local extreme qualities move toward a couple of fish keeping an eye on global value, which brings about AF escaping from the local extreme values.

Mathematical equation and Representation in DFSO
Discrete solution: $M_{in} = \{\text{rand}(P_i, \text{ul} - \text{lb}) + \|\|$ (11)

Prey:
$M_j = M_j + \text{visual rand}$

Swarm:
$M_j^{(t+1)} = M_j^{(t)} + \frac{M_c - M_j^{(t)}}{||M_c - M_j^{(t)}||} \cdot \text{step.rand}$

Follow: Shown in equation (15)

Leave and Move: $M_j^{(t+1)} = M_j^{(t)} + \alpha \cdot \text{rand}$

Above the mathematical equation $P_i$ as processing time and $j*M$ problem, $ul$ and $ll$ as the upper limit and lower limit, $M_c$ center position.
K. Hybrid optimization for FJSSP

IV. IMPLEMENTATION OF RESULTS ANALYSIS

This proposed hybrid optimization of FJSSP, implemented in MATLAB 2016a with the system configuration, i5 processors with 4GB RAM. The benchmark data collected from http://people.brunel.ac.uk/~mastjjb/jeb/orlib/, it’s having many databases’ Validation and the results are discussed in this section. Similarly the other two objectives of FJSSP. For every instance, the minimum workload value noted as an optimal one. It is used definite makespan time to show how this study works properly and how this study result is better than the other studies result in this method.

Therefore, the Gantt char represents the scheduling comparing to the solution in fig 5 and table 3. It also shows the results of MKO7 (20 * 10), processing time like 25, 55, 76 and 58. The number of solution to the problem of 8 by 8 is less, and it also induced the probability of multi – optimal solution is increased. The capacity to avoid local minimum solutions is expanded with the diversity rate increasing. These job elements are processed to provide the minimum makespan time that is added to their corresponding job sequence input time. The number of the machine operated for the problem size varies, and find the processing time.

According to the request of operations, various jobs created the particle position. A particle position is comparing to a machine task all considered operation in light of an alternate priority level relating to an alternative machine

Fig. 4. Sample Convergence Graph (Time)

Fish swim promptly in water; truth be told; they are found for food or partners in a wider range. In water, the leave procedure was stopped someplace and the result of AFs behavior will be the same, the distinction between objectives values (food concentration, FC) end up littler within a few cycles. It may fall into local extreme change the parameters haphazardly to the still states for leaping outflow state (28). These operations of anomalous are presented in condition (16) Swarm makes a fish limit in local extreme qualities deploy toward a couple of fish focus on the global value that brings about AF emerging from the local extreme values.

The rate of convergence of the benchmark problem (MKO2) presented in figure 5 with three optimization models which goals to attain the optimal solutions. One to five experiments for various iteration from the range and it is invented that the vast majority of the problem gave great outcomes. During repeated iterations, the convergence graph can situate the optimal fitness value. The close optimal solution with relatively small makes on value is more possible. The hybrid optimization achieved by the optimal fitness value, for instance, minimal value contrasted with CSO and DFSO in the proposed method.

(a) Makespan Time
Fig. 5. Comparative Analysis

In fig 4 (a), (b) and (c) presented the makespan time, total work load and maximum workload comparative analysis. Hybrid optimization, CSO, DFSO, HCPSEO [15], PSO [16] obtained non-dominated solutions and the compared algorithm dominated the DABC [18] algorithm. (a) Shows the makespan time, it’s compared to all optimization with all instance averagely minimum in our proposed work.

Fig. 6. Gantt chart

The scheduling is comparing to the Solution, therefore represented by the Gantt chart in figure 5 and table 3. Here showing the results of MK07 (20x10), processing time like 25, 55, 76 and 58. There are a number of solutions to the problem of 8 by 8 is less, it also induced the probability of multi-optimal solution to increase. The rate increases with the diversity, the capacity to avoid local minimum solutions are expanded. These job sequences are processed to give the minimum makespan time, which is added to their corresponding job sequence input time. Here, if the problem size varies, the processing time is also found to vary based on the number of the machine operated. Then the particle position can be created stochastically as per the request of operations of various jobs. In light of an alternate priority level relating to an alternate machine, a particle location is comparing to a machine task all considered operations.

V. CONCLUSION

The present paper enhances the flexible job shop scheduling problem with the different benchmark problems. The real procedure is to minimize the makespan with multi-objectives and to effectively overcome the infeasible solutions the objective is developed. So, the guess of the global optimum turns out to be more precisely proportional to the number of iterations. The other one is sometimes local optimum can be happened rather than the global optimum. Multi hybrid optimization investigated with various goals for more than 20 cases. The finest accuracy accomplishes in CSO and DFSO, moreover decreases the makespan time for all machines. The work can be additionally extended by considering the various exposures which are practiced in a genuine FJSSP problem. In future research work, might be lessened makespan time with different planning procedures.

APPENDIX

I. OBJECTIVE FUNCTION RESULTS OF BENCHMARK PROBLEMS

<table>
<thead>
<tr>
<th>Problem</th>
<th>CSO</th>
<th>DFSO</th>
<th>Hybrid</th>
<th>HCPSEO(20)</th>
<th>PSO(21)</th>
<th>DABC(25)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KA02</td>
<td>450</td>
<td>460</td>
<td>440</td>
<td>430</td>
<td>420</td>
<td>410</td>
</tr>
<tr>
<td>KA04</td>
<td>850</td>
<td>860</td>
<td>840</td>
<td>830</td>
<td>820</td>
<td>810</td>
</tr>
<tr>
<td>KA05</td>
<td>15X10</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>KA06</td>
<td>10X6</td>
<td>85</td>
<td>84</td>
<td>83</td>
<td>82</td>
<td>81</td>
</tr>
<tr>
<td>MK01</td>
<td>10X6</td>
<td>72</td>
<td>71</td>
<td>70</td>
<td>69</td>
<td>68</td>
</tr>
<tr>
<td>MK02</td>
<td>15X5</td>
<td>78</td>
<td>77</td>
<td>76</td>
<td>75</td>
<td>74</td>
</tr>
<tr>
<td>MK03</td>
<td>20X2</td>
<td>62</td>
<td>61</td>
<td>60</td>
<td>59</td>
<td>58</td>
</tr>
<tr>
<td>MK07</td>
<td>20X10</td>
<td>53</td>
<td>52</td>
<td>51</td>
<td>50</td>
<td>49</td>
</tr>
<tr>
<td>Average</td>
<td>65</td>
<td>64</td>
<td>63</td>
<td>62</td>
<td>61</td>
<td>60</td>
</tr>
</tbody>
</table>

II. OPERATION TIME FOR MK07 (20X10)

<table>
<thead>
<tr>
<th>Job</th>
<th>Operation Time</th>
<th>Available Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>96</td>
<td>122</td>
</tr>
<tr>
<td>A2</td>
<td>106</td>
<td>132</td>
</tr>
<tr>
<td>A3</td>
<td>116</td>
<td>142</td>
</tr>
<tr>
<td>A4</td>
<td>126</td>
<td>152</td>
</tr>
<tr>
<td>A5</td>
<td>136</td>
<td>162</td>
</tr>
<tr>
<td>B1</td>
<td>97</td>
<td>123</td>
</tr>
<tr>
<td>B2</td>
<td>107</td>
<td>133</td>
</tr>
<tr>
<td>B3</td>
<td>117</td>
<td>143</td>
</tr>
<tr>
<td>B4</td>
<td>127</td>
<td>153</td>
</tr>
<tr>
<td>B5</td>
<td>137</td>
<td>163</td>
</tr>
<tr>
<td>C1</td>
<td>98</td>
<td>124</td>
</tr>
<tr>
<td>C2</td>
<td>108</td>
<td>134</td>
</tr>
<tr>
<td>C3</td>
<td>118</td>
<td>144</td>
</tr>
<tr>
<td>C4</td>
<td>128</td>
<td>154</td>
</tr>
<tr>
<td>C5</td>
<td>138</td>
<td>164</td>
</tr>
<tr>
<td>D1</td>
<td>99</td>
<td>125</td>
</tr>
<tr>
<td>D2</td>
<td>109</td>
<td>135</td>
</tr>
<tr>
<td>D3</td>
<td>119</td>
<td>145</td>
</tr>
<tr>
<td>D4</td>
<td>129</td>
<td>155</td>
</tr>
<tr>
<td>D5</td>
<td>139</td>
<td>165</td>
</tr>
</tbody>
</table>
REFERENCES


AUTHORS PROFILE

S. Kavitha, received her Bachelor degree in Mechanical Engineering from Bharathiar University, Coimbatore, India and Masters in CAD from Sathyabama University, Chennai, India. She is currently pursuing her PhD in Kalasalingam University, Tamilnadu, India. She is also working as an Assistant Professor in the Department of Mechanical Engineering, Kalasalingam University. Her research interests include operations management and scheduling.

Dr. P. Venkumar, is currently a Senior Professor in Department of Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnankovil, Tamilnadu, India. He received his B.E., and M.E., degrees in Thigagrajgar College of Engineering, Madurai, Tamilnadu, India, in 1994 and 1997, respectively and Ph.D. degree in Manonmaniam Sundaranar University, India in Industrial Engineering in 2006. His main areas of research interest are Manufacturing system optimization, Supply chain management and Smart manufacturing.