Multi-objective flexible flow shop scheduling with unexpected arrivals of new jobs

Shahram Entezari*; Saeedeh Gholami
Department of industrial engineering, K.N.Toosi of Technology, Iran
Sh.entezari@kntu.ac.ir

Abstract
This paper considers a two-stage flexible flow shop scheduling problem with unexpected arrival of new jobs. In this study, a two-step approach is used. In the first step, an initial schedule is obtained considering makespan as objective function. After that the initially schedule is applied, now assume that a new job arrives during the execution of the initial schedule. In the second step, we propose three criteria as a measure based on a classical objective and performance measures. This Measure contain of makespan, stability and Variation of completion times. Computational results indicate the promising performance of the proposed approach in scheduling unexpected arrival jobs.

Keywords: flexible flow shop scheduling, uncertainty, multi-objective, stability

1. Introduction

A flexible flowshop (FFS) is a generalization of the flow shop and the parallel processor environments. A flexible flow shop is alternatively called a hybrid flow shop or multiprocessor flowshop. In the most general setting of a flexible flowshop environment, there are multiple stages (S stages), each of which consists of m(s) (s = 1, 2, 3,...,S) parallel processors). A schematic representation of a flexible flowshop environment is given in Figure 1.3. The processors in each stage may be identical, uniform, or unrelated. Machines are uniform if the time to process a job on any machine is a constant ratio of its processing time on other machines. In other words, uniform machines are identical processors that do not have equal speeds. Unrelated machines are machines for which the time to process a job on any machine has no particular relationship of its processing time on any other machine (Cheng & Sin, 1990). In a FFS environment, each job is processed first at stage 1, then at stage 2, and so on. Normally, a job requires only one machine at each stage and any machine can process any job.

In most real-world manufacturing environments, production scheduling is a reactive process subject to a variety of uncertainties, referred to as unexpected events, such as machine breakdown, stochastic processing times, rush order, job cancellation, and change of due date, etc. In order to moderate the negative effect of uncertainties, the unexpected events and their effect have to be considered during problem-solving. A significant number of successful implementations of real-world scheduling have been found in many manufacturing systems, including single machine systems, parallel machine systems, flow shops, job shops, open shops, and flexible manufacturing systems.

Liu et al. (2007) presented a new method considering both robustness and stability for single machine scheduling under machine breakdown. Allaoui and Artiba (2004) integrated a flexible simulation model with SA heuristic to address FFS scheduling with maintenance constraints. The simulation model was employed to evaluate solutions generated by SA.

Kianfar et al. (2009) developed four new dispatching rules to solve the FFS scheduling problems considering dynamic job arrivals. The simulation results indicated the better performance of the proposed approaches when minimizing the sum of rejection and tardiness costs of jobs. Kia et al. (2010) studied a dynamic FFL problem with sequence-dependent setup and random job arrivals. They established a discrete-event simulation model to evaluate the performance of seven dispatching rules and some proposed heuristics.

The rest of the paper is organized as follows: section 2 explains the problem. Section 3 provides a description of scheduling mathematical model for determine initial schedule. Section 4 presents the new reactive approach
considering arrival of new jobs. Numerical results are given in section 5. Finally, in Section 6 some conclusions on this study are given.

2. Problem description

Flexible Flow Shop Scheduling is extended to flow Shop Scheduling. Figure (1) indicates a general framework of Flexible Flow Shop Scheduling that contains m stages and variable number of machines at each stage. In this paper we propose flexible flow shop with unexpected arrivals of new jobs, first an initial sequence of jobs should be determined. It is initially supposed that there is no random disruption in the system. So, in the first, an initial solution was calculated for scheduling by a scenario-based robust mathematical model. After an initially schedule is specified, the machines start to process the jobs according to this design. However, during the processing of these jobs some unexpected new jobs may arrive at the system. We proposed three criteria to decrease impact of disruption.

3. Determine the initial schedule

In this section, a mixed integer linear programming formulation is provided for aforementioned flexible flow shop problem based on the mathematical model proposed in (Guinet et al., 1996) which the makespan is considered as a classical objective function. The following model is used to determine the initial sequence of jobs in the state of certainty. At first, the parameters, indices and variables are defined as below:

Indices

\[
I = \text{index for machines } \{1, \ldots, m_l\}
\]

\[
j, h, r = \text{index for jobs } \{1, \ldots, n\}
\]

\[
L = \text{index for stages } \{1, 2\}
\]

\[
S = \text{index for scenarios } \{1, \ldots, s\}
\]

Parameters

\[
\eta = \text{number of machines in machine center l}
\]

\[
t_j^s = \text{processing time for job j in stage l under scenario s}
\]

\[
p = \text{the probability of happening of scenario s}
\]

\[
M = \text{a large number}
\]

Variables

\[
\begin{cases}
1 & \text{If job j is processed directly after job r on machine i in stage l} \\
0 & \text{Otherwise}
\end{cases}
\]

\[
\begin{cases}
1 & \text{If job j is the first job to be processed on machine i in stage l} \\
0 & \text{Otherwise}
\end{cases}
\]

\[
\begin{cases}
1 & \text{If job i is the last job to be processed on machine i in stage l} \\
0 & \text{Otherwise}
\end{cases}
\]

\[
\begin{cases}
1 & \text{completion time of job j in machine center l} \\
0 & \text{Otherwise}
\end{cases}
\]

\[
C_{jt} = \text{completion time of job j in machine center l after disruption}
\]

\[
C = \text{maximum completion time of the jobs}
\]
The model established with makespan objective function is as follows:

\[
\text{MinZ} = C_{\text{max}}
\]  

Subject To :

1. \[\sum_{r=0}^{n} \sum_{j=0}^{m} x_{gil} = 1, \forall j = 1, ..., n, L = 1, ..., l\]  
2. \[\sum_{j=0}^{n} x_{gil} \leq 1, \forall i = 0, ..., n, i = 1, ..., m, L = 1, ..., l\]  
3. \[\sum_{r=0}^{n} x_{ijkl} - \sum_{j=0}^{n} x_{hjil} = 0, \forall h = 1, ..., n, h = 1, ..., m, L = 1, ..., l\]  
4. \[c_{jl} + M \left(1 - \sum_{k=1}^{m} x_{jkkl}\right) \geq c_{jl} + \sum_{i=1}^{m} x_{gil} t_{jl}, \forall r, j = 1, ..., n, l = 1, ..., L\]  
5. \[c_{jl} \geq c_{j, j+1} + t_{jl}, \forall j = 0, ..., n, l = 1, ..., L\]  
6. \[c_{jl} \leq C_{\text{max}}, \forall j = 1, ..., n, l = 1, ..., L\]  
7. \[x_{gil} \in \{0, 1\}, \forall r, j = 1, ..., n, i = 1, ..., m, l = 1, ..., L\]  
8. \[c_{jl} \geq 0, \forall j = 1, ..., n, l = 1, ..., L\]

The objective function (1) states that we seek to minimize makespan. Constraint (2) states that every job is processed once in each stage. Constraint (3) specifies that every job is accompanied by at most one succeeding job. Constraint (4) ensures that each job has a job predecessor and a job successor on its machine. Constraint (5) calculates the job completion times and forbids a job to be both the predecessor and the successor of the same job. Constraint (6) guarantees that each job must be processed in series in the stages. Constraint (7) calculates the makespan.

4. Reactive approach

An initial solution has been achieved for existing jobs. Now, assume that a new job arrives during the execution of the initial schedule. Considering the new job, our proposed method adopts an appropriate reactive action in order to determine its order in the initial sequence.

4.1. Scheduling effectiveness

This criterion shows the amount of optimization for a schedule. This criterion is measured by the classical objective “makespan”. We should mention that because of the disruptions in the system, the real completion time of affected jobs may change. We define the variable \(C_{jl}\) as the completion time of job \(j\) on stage \(l\). So \(C_{\text{max}}\) is the makespan in scheduling that explains the scheduling effectiveness.

\[f 1 = C_{\text{max}}\]

4.2. Stability

Recall that a stable schedule is one that should not deviate much from the initial schedule. The deviation is generally measured in terms of the differences between the job completion times in the initial and realized schedules. If a schedule encounters a degree of disruption that does not differ much relative to the original schedule, it is called stable. Hence, a typical stability measure is a non-decreasing function of the deviation of
job completion times. We use stability measure that was already available in the literature. The stability is defined as follows (Al-Hinai and ElMekkawy, 2011).

\[ f_2 = \sum_{l} \sum_{j} |C_{jl}' - C_{jl}| \]

But since this term is non-linear, the equation is transformed to:

\[ y_{jl} = C_{jl}' - C_{jl} \]
\[ f_2 = \sum_{l} \sum_{j} y_{jl} \]

\[ 1. \quad y_{jl} \geq C_{jl}' - C_{jl} \]
\[ -y_{jl} \leq C_{jl}' - C_{jl} \]

4.3. Variation of completion times

We defined this criterion as the deviation in performance of the real schedule from the initial schedule. This measure specifies the closeness of the performance of the real schedule to the initial schedule. In fact, the robustness associated with the changes created in the objective function following the disruptions. If the performance of a schedule, in dealing with disruptions, does not become too weak and deficient, it is called a robust schedule.

The important point to mention is that in the first step, in order to determine the initial schedule, the processing times were estimated as scenarios. In other words, the setΩ of scenarios was considered to define the uncertain processing times and based on those, the initially robust schedule was determined. The presumption is that in reality, only one of these scenarios will occur in the future. Therefore when the manufacturing process begins, it is absolutely certain which scenario has occurred. Considering this notion, suppose that 8,, ∈ is the scenario which has really occurred during the processing of jobs. In such a case, the real completion times are specified based on scenario 8,, ∈ . Based on this matter and the definition of robustness of schedule we define the robustness as follows:

\[ f_3 = \sum_{j} |C_{jl} - \bar{C}| \]

Since this equation is non-linear, the equation is converted to:

\[ z_{j} = C_{jl} - \bar{C} \]

\[ 1. \quad \bar{C} = \frac{\sum_j C_{jl}}{n} \]

\[ f_3 = \sum_j z_j \]

\[ 1. \quad z_{j} \geq C_{jl} - \bar{C} \]
\[ -z_{j} \leq C_{jl}' - \bar{C} \]

4.4. Objective function

Objective function of this problem is defined as a single objective based on these three functions. The terms in (12), (13) and (14) will be normalized to enable a reasonable comparison. The stable modified function is defined as follows:

\[ MinZ = w_1 \left( \frac{f_3 - f_3^+}{f_3^- - f_3^+} \right) + w_2 \left( \frac{f_2 - f_2^+}{f_2^- - f_2^+} \right) + w_3 \left( \frac{f_1 - f_1^+}{f_1^- - f_1^+} \right) \]

Where \( w_1, w_2, w_3 \) are the weight coefficients of the weighted makespan, stability and Variation of completion times in the objective function. The \( f^- \) and \( f^+ \) are the worst and the best values for each
function.
To obtain $f^-$ and $f^+$ first, considering every criterion to solve the problem alone that according to three criteria, each problem should be solved three times. The results are shown as a $3 \times 3$ matrix in equation (3-39).

$$
\begin{bmatrix}
    f_1 & f_2 & f_3 \\
    f_1^+ & a_{11} & a_{12} & a_{13} \\
    f_2^+ & a_{21} & a_{22} & a_{23} \\
    f_3^+ & a_{31} & a_{32} & a_{33}
\end{bmatrix}
$$

In matrix (3-39) the first row corresponds to the case where considering $f_1$ model of the problem is solved. The second and third rows correspond to the state that by considering $f_2, f_3$ the problem are solved. Values of $f^-$ and $f^+$ are calculated as follows.

$$
f_1^+ = \min\{a_{11}, a_{21}, a_{31}\} \\
f_2^+ = \min\{a_{12}, a_{22}, a_{32}\} \\
f_3^+ = \min\{a_{13}, a_{23}, a_{33}\} \\
f_1^- = \max\{a_{11}, a_{21}, a_{31}\} \\
f_2^- = \max\{a_{12}, a_{22}, a_{32}\} \\
f_3^- = \min\{a_{13}, a_{23}, a_{33}\}
$$

The proposed model is:

$$
\text{Min} Z = w_1 \left( \frac{f_1 - f_1^+}{f_1^- - f_1^+} \right) + w_2 \left( \frac{f_2 - f_2^+}{f_2^- - f_2^+} \right) + w_3 \left( \frac{f_3 - f_3^+}{f_3^- - f_3^+} \right)
$$

(10)

$$
f_1 = C_{\text{max}}$$

(11)

$$
f_2 = \sum_j \sum_{r_j} y_{rj}$$

(12)

$$
f_3 = \sum_j z_{rj}$$

(13)

Subject To:

$$
\sum_{r=0}^{n} \sum_{r \neq j}^{n} x_{rji} = 1, \forall j = 1, ..., n, L = 1, ..., l
$$

(14)

$$
\sum_{j=0}^{n} x_{rji} \leq 1, \forall i = 0, ..., n, i = 1, ..., m_j, L = 1, ..., l
$$

(15)

$$
\sum_{r=0}^{n} x_{rki} - \sum_{h=0}^{n} x_{hjl} = 0, \forall h = 1, ..., n, i = 1, ..., m_j, L = 1, ..., l
$$

(16)

$$
c_{rj}^+ + M \left( 1 - \sum_{k=1}^{m_j} x_{rkl} \right) \geq c_{rj}^+ + \sum_{r=1}^{m_j} x_{rjl} t_{rj}^+, \forall r, j = 1, ..., n, l = 1, ..., L, S = 1, ..., s
$$

(17)

$$
c_{rj}^+ \geq c_{rj}^+ + t_{rj}^+, \forall j = 0, ..., n, l = 1, ..., L, S = 1, ..., s
$$

(18)

$$
c_{rj}^+ \leq C_{\text{max}}, \forall j = 1, ..., n, l = 1, ..., L, S = 1, ..., s
$$

(19)

$$
y_{rj} \geq C_j^+ - C_{rj}
$$

(20)
The objective function (19) states that we seek to minimize makespan, stability and Variation of completion times.

5. Numerical example

In this section we assume that there are 14 jobs that must be processed in two stages and each stage has 3 machines. Processing times of each job is differently on each machine in each stage. In fact, there are unrelated parallel machines in each stage. But jobs must be processed at each stage, only on a machine and it is no different for each machine which it is processed. The first model with regard to the makespan, as the objective function can be solved using GAMS software. The result and the sequence of any jobs on machines at any stage are shown in the table ().

Table 1. Processing time for job j on machine i at stage l.

<table>
<thead>
<tr>
<th>stage</th>
<th>machine</th>
<th>job 1</th>
<th>job 2</th>
<th>job 3</th>
<th>job 4</th>
<th>job 5</th>
<th>job 6</th>
<th>job 7</th>
<th>job 8</th>
<th>job 9</th>
<th><code>job 10</code></th>
<th>job 11</th>
<th>job 12</th>
<th>job 13</th>
<th>job 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>33</td>
<td>48</td>
<td>27</td>
<td>32</td>
<td>36</td>
<td>38</td>
<td>47</td>
<td>23</td>
<td>35</td>
<td>41</td>
<td>25</td>
<td>49</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>47</td>
<td>26</td>
<td>34</td>
<td>36</td>
<td>22</td>
<td>39</td>
<td>33</td>
<td>44</td>
<td>30</td>
<td>32</td>
<td>23</td>
<td>21</td>
<td>43</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>48</td>
<td>50</td>
<td>33</td>
<td>30</td>
<td>41</td>
<td>24</td>
<td>28</td>
<td>34</td>
<td>21</td>
<td>26</td>
<td>49</td>
<td>46</td>
<td>38</td>
<td>27</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>29</td>
<td>21</td>
<td>39</td>
<td>37</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>24</td>
<td>32</td>
<td>35</td>
<td>38</td>
<td>41</td>
<td>33</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>39</td>
<td>37</td>
<td>27</td>
<td>26</td>
<td>36</td>
<td>30</td>
<td>28</td>
<td>46</td>
<td>22</td>
<td>29</td>
<td>48</td>
<td>40</td>
<td>44</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>22</td>
<td>21</td>
<td>50</td>
<td>40</td>
<td>24</td>
<td>37</td>
<td>33</td>
<td>25</td>
<td>48</td>
<td>28</td>
<td>46</td>
<td>31</td>
<td>38</td>
<td>32</td>
</tr>
</tbody>
</table>

Table 2. The initial schedule based on first step with Cmax=187 and runtime=1627(s)

<table>
<thead>
<tr>
<th>stage</th>
<th>machine</th>
<th>Sequence of jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3 5 11 4</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>2 14 12 9 13</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>10 6 7 8 1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>14 11 8 13</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3 10 7 9 4</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6 5 12 2 1</td>
</tr>
</tbody>
</table>
After that the initially schedule is applied, now assume that a new job arrives during the execution of the initial schedule. No predictive information exists for this job. This unexpected arrival is an issue related to online scheduling in the dynamic environment of this system. Our proposed method implements an applicable reactive action considering the new job that determine its order in the initial sequence. Assume that the new job arrives after the first 8 jobs process. 6 remaining jobs and new arrival job should be scheduled regard to makespan, stability and variation of completion times.

The mathematical model of this study is a multi-objective programming that normalized and solved by Compromise Programming (CP) method. First, $f_i^+$ and $f_i^-$ should be obtained by Compromise Programming method. Results of CP method are shown in table (3).

**Table 3. Results of CP method**

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1^*$ ($w_1 = 1, w_2 = 0, w_3 = 0$)</td>
<td>107.000</td>
<td>444.000</td>
<td>338.143</td>
</tr>
<tr>
<td>$F_2^*$ ($w_1 = 0, w_2 = 1, w_3 = 0$)</td>
<td>179.000</td>
<td>92.000</td>
<td>321.571</td>
</tr>
<tr>
<td>$F_3^*$ ($w_1 = 0, w_2 = 0, w_3 = 1$)</td>
<td>156.000</td>
<td>715.000</td>
<td>150.000</td>
</tr>
<tr>
<td>$f_i^+$</td>
<td>107.000</td>
<td>92.000</td>
<td>150.000</td>
</tr>
<tr>
<td>$f_i^-$</td>
<td>179.000</td>
<td>715.000</td>
<td>338.143</td>
</tr>
</tbody>
</table>

In the following the ten problems is solved with different weights. The results of the rescheduling for ten test problem are shown in Table 4.

**Table 4. Objective function and CPU time obtained by the proposed method.**

<table>
<thead>
<tr>
<th>Problem number</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>Obj. value</th>
<th>CPU. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.8</td>
<td>141.143</td>
<td>286.857</td>
<td>150.000</td>
<td>0.079</td>
<td>851.634</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
<td>139.000</td>
<td>162.000</td>
<td>262.857</td>
<td>0.228</td>
<td>27.638</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>0.6</td>
<td>0.1</td>
<td>139.000</td>
<td>164.000</td>
<td>262.857</td>
<td>0.263</td>
<td>58.111</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td>128.143</td>
<td>289.000</td>
<td>194.000</td>
<td>0.282</td>
<td>113.205</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.1</td>
<td>115.000</td>
<td>319.000</td>
<td>259.143</td>
<td>0.259</td>
<td>166.821</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>0.1</td>
<td>0.3</td>
<td>123.000</td>
<td>282.000</td>
<td>214.571</td>
<td>0.267</td>
<td>119.154</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.2</td>
<td>0.1</td>
<td>109.000</td>
<td>426.000</td>
<td>317.143</td>
<td>0.216</td>
<td>213.934</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>109.000</td>
<td>419.000</td>
<td>346.143</td>
<td>0.179</td>
<td>317.926</td>
</tr>
<tr>
<td>9</td>
<td>0.9</td>
<td>0.05</td>
<td>0.05</td>
<td>118.000</td>
<td>614.000</td>
<td>279.143</td>
<td>0.214</td>
<td>565.214</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>107.000</td>
<td>493.000</td>
<td>363.143</td>
<td>0.000</td>
<td>701.236</td>
</tr>
</tbody>
</table>

By changing the weights of the criteria defined 10 problems that result is shown in a table ( ). To solve every problem, a solution to the problem is obtained which is a Pareto point. Point 3 dominates by point 2 then there are 9 non-dominatesolution that is shown in the ( )..
According to Table 4 and Figure 1 the objective function of the first and third criteria is more sensitive than the second criterion. For example, with regard to the problem number 3 and 4 with respect to decrease of the first and third criterion weight and increase of the second criterion weight, objective function value decreases.

With regard to problem 1 and 8, we find that the sensitivity of the objective function value of the third criterion is more than first.

According to the figure 2 and results in Table 4 it can be seen the runtime of this model is very sensitive to the first and third criteria and greatly increased with increasing weights of these two criteria, but the second criterion has little sensitivity than other two criteria. Then runtime of model is to be extremely sensitive to the first and third criterion.

To illustrate this situation several issues are systematically defined that results of which are shown in the following tables.

Table 5. Solving various problems to understand the trend of runtime and objective function value

<table>
<thead>
<tr>
<th>Weight Scheme</th>
<th>Objective Function Value</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>B. weight of the first measure fixed and equal to 0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A. weight of the third measure fixed and equal to 0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Problem number</td>
<td>$w_1$</td>
<td>$w_2$</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>0.6</td>
</tr>
<tr>
<td>7</td>
<td>0.1</td>
<td>0.7</td>
</tr>
<tr>
<td>8</td>
<td>0.1</td>
<td>0.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem number</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>Obj. value</th>
<th>CPU. Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.8</td>
<td>0.1</td>
<td>0.174</td>
<td>30.699</td>
</tr>
<tr>
<td>2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
<td>0.228</td>
<td>27.019</td>
</tr>
<tr>
<td>3</td>
<td>0.3</td>
<td>0.6</td>
<td>0.1</td>
<td>0.263</td>
<td>57.145</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>0.5</td>
<td>0.1</td>
<td>0.276</td>
<td>97.173</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.1</td>
<td>0.259</td>
<td>166.069</td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
<td>0.234</td>
<td>168.389</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.2</td>
<td>0.1</td>
<td>0.216</td>
<td>215.635</td>
</tr>
<tr>
<td>8</td>
<td>0.8</td>
<td>0.1</td>
<td>0.1</td>
<td>0.179</td>
<td>317.926</td>
</tr>
</tbody>
</table>

Due to problems A and figure 3 observed when the first criterion weight increases the runtime increases exponentially and the objective function value is increases first and then decreases.

According to set problem B when the third criterion weight reduced the objective function value is increases and runtime decrease exponentially.
6. Conclusion

In most real-world manufacturing environments, production scheduling is a reactive process subject to a variety of uncertainties, referred to as unexpected events, such as machine breakdown, stochastic processing times, rush order, job cancellation, and change of due date, etc. The occurrence of uncertainties affects the system status and causes deviations from the planned schedules. To deal with these disruptions, stable solutions should be obtained for the scheduling problems. Computational results presented that this method was much more effective compared with ordinary scheduling methods that operate on the basis of makespan alone. The experimental results of these scheduling problems are encouraging, which shows that this method is an effective methodology for solving dynamic flexible flow shop scheduling problems by unexpected arrival of new jobs. For future research, this problem can be considered for other shop floor scheduling problems, and other classical objectives can be used to evaluate this method.

References


