KAOHSIUNG

# APIEMS & CIIE 2007

Proceedings of The 8th Asia Pacific Industrial Engineering & Management System and 2007 Chinese Institute of Industrial Engineers Conference

## Service Value Innovation and Challenges

-December 9 - 12, 2007, The Ambassador Hotel, Kaohsiung, Taiwan-

***Sponsor***

Asia Pacific Industrial Engineering and Management Society

Chinese Institute of Industrial Engineers

***Conference Organizer***

Department of Industrial Engineering and Management

National Yunlin University of Science and Technology

# Fuzzy dispatching rules for flexible flow shop problems with unrelated parallel machines for a continuous fuzzy domain

**Jitti Jungwattanakit [1*], Manop Reodecha [2], Paveena Chaovalitwongse [3], and Frank Werner [4]**

[1, 2, 3] Department of Industrial Engineering, Faculty of Engineering
Chulalongkorn University, Bangkok 10330 Thailand
Corresponding author's e-mail: jitti.j@student.chula.ac.th

[4] Faculty of Mathematics
Otto-von-Guericke-University, P.O. Box 4120, D-39016 Magdeburg, Germany
Phone: +49-391-6712025, fax: +49-391-6711171
e-mail: frank.werner@mathematik.uni-magdeburg.de

**Abstract:** A flexible flow shop problem can be considered as a generalization of a pure flow shop problem in which the jobs have to go through the *k* stages in the same order. We consider a flexible flow shop problem with unrelated machines and setup times, where the processing times depend on the chosen machine of each stage and setup times are sequence-dependent. While for classical problems the processing times for each job are assumed to be known exactly, in many real-world situations processing times vary dynamically due to occasional machine breakdowns, levels of operator's skills, operating faults, or other human factors. In this paper, fuzzy concepts are used for the dispatching rules (i.e. LPT, SPT and ERD) for managing uncertain scheduling. Given a set of jobs together with a membership function for the standard processing times, the fuzzy dispatching rules construct a solution by means of a membership function for the final makespan. Computational experiments show that among the fuzzy SPT algorithms, a fuzzy SPT algorithm using the average values of the fuzzy operating times at the first stage (denoted by $FSPT^1$) gives the best solutions whereas among the fuzzy LPT algorithms, a fuzzy LPT algorithm using the average values of the fuzzy operating times at the last stage (denoted by $FLPT^k$) gives the best solutions for both small- and large-size test problems. Comparing all fuzzy dispatching algorithms considered in this paper, the results show that the $FLPT^k$ algorithm outperforms the others. In particular, the $FLPT^k$ algorithm gives a deviation from the optimal makespan value of about five percent for small-size test problems. The proposed algorithms provide a more flexible method of scheduling jobs than conventional scheduling methods.

**Keywords:** Flexible Flow Shop, Unrelated Parallel Machines, Fuzzy Sets, Dispatching Rules.

## 1. INTRODUCTION

In recent years, new products and changes in existing products are occurring with an increasing rate, causing pressure on companies to reduce lead times and to maintain a high on-time delivery performance. Thus, an effective scheduling of jobs becomes a key to achieving their goals. This paper is primarily concerned with a scheduling problem occurring in the production industries. They are established as multi-stage production flow shop facilities where a production stage may be made up of parallel machines. This is known as flexible flow shop or hybrid flow shop environment, i.e. it is a generalization of the classical flow shop model. There are *k* stages and some stages may have only one machine, but at least one stage must have multiple machines, and all jobs have to pass through the *k* stages in the same order. In such industries, it is common to find newer or more modern machines running side by side with older and less efficient machines. The older machines may perform the same operations as the newer ones, but would generally require a longer operating time for the same operation. Moreover, it may be possible that speeds of machines are dependent on the jobs, as well. Such a problem is called a flexible flow shop problem with unrelated parallel machines (see Jungwattanakit, Reodecha, Chaovalitwongse, and Werner [1-3]). In [1-3], it has been found that the SPT (Shortest Processing Time), LPT (Longest Processing Time) and ERD (Earliest Release Date first) algorithms are good dispatching rules for the makespan problem. The latter work dealt with the situation that the processing times for each job are exactly given as deterministic values. However, in many real-world applications, processing times may vary dynamically due to occasional machine breakdowns, levels of operator's skills, operating faults or other human factors. The estimated processing times are not precisely known. It is therefore reasonable to consider them as uncertain variables. Consequently, several concepts such as fuzzy set theory, probability theory, DEMPSTER/SHAFER theory, sensitivity analysis, and others, have been used to take into account the uncertainties [4].

In this paper, we will treat uncertainty by using fuzzy set theory because of its simplicity and similarity to human reasoning [5]. Such a theory has been applied to many areas such as inventory control [6] and scheduling [7]. We apply fuzzy dispatching (in particular, LPT, SPT and ERD) algorithms to the problem under consideration. Given a set of jobs, each of which has its membership function for the standard processing times, a scheduling result with a membership function for the final completion time is generated.

The remainder of this paper is as follows. In Section 2, the problem under consideration is described. Section 3 presents fuzzy dispatching algorithms. A numerical example is discussed in Section 4. Computational results are discussed in Section 5 and some conclusions are given in Section 6.

## 2. PROBLEM DESCRIPTIONS

Flexible flow shop problems can be described as follows. There is a set $J = \{1,\dots, j,\dots, n\}$ of $n$ independent jobs which have to be processed, and the processing system is defined by a set $O = \{1,\dots, t,\dots, k\}$ of $k$ processing stages. At each stage $t$, $t \in O$, there is a set $M^t = \{1,\dots, i,\dots, m^t\}$ of $m^t$ unrelated machines. Each job $j$, $j \in J$, has its release date $r_j \geq 0$ and a due date $d_j \geq 0$. Due to the unrelated machines, the processing time $p_{ij}^t$ of job $j$ on machine $i$ at stage $t$ is equal to $ps_j^t / v_{ij}^t$, where $ps_j^t$ is the standard processing time of job $j$ at stage $t$, and $v_{ij}^t$ is the relative speed of job $j$ which is processed by the machine $i$ of stage $t$. However, since the standard processing time is uncertain, it is represented by a fuzzy number. Consequently, each job has a fuzzy standard processing time $\underset{\sim}{ps}_j^t$ for every stage $t$, $t \in O$.

There are processing restrictions of the jobs as follows: (1) jobs are processed without preemptions on any machine; (2) every machine can process only one operation at a time; (3) operations of a job have to be realized sequentially, without overlapping between the stages; (4) job splitting is not permitted.

Setup times considered in this problem are classified into two types, namely a machine-dependent setup time and a sequence-dependent setup time. A setup time of a job is machine-dependent if it depends on the machine to which the job is assigned. It is assumed to occur only when the job is the first job assigned to the machine. $ch_{ij}^t$ denotes the machine-dependent setup time (or changeover time) of job $j$ if job $j$ is the first job assigned to machine $i$ at stage $t$. A sequence-dependent setup time is considered between successive jobs on the same machine. A setup time of a job on a machine is sequence-dependent if it depends on the job just completed on that machine. $s_{lj}^t$ denotes the time needed to changeover from job $l$ to job $j$ at stage $t$, where job $l$ is processed directly before job $j$ on the same machine. All setup times are known and constant. Moreover, there is given a non-negative machine availability time for any machine of a particular stage.

The objective is to minimize the fuzzy makespan $\underset{\sim}{C}_{max}$ which is equivalent to the fuzzy completion time of the last job leaving the system.

## 3. FUZZY SCHEDULING ALGORITHMS

In this section, fuzzy set theory is used for the LPT, SPT and ERD algorithms to schedule the jobs with uncertain standard processing times. Given a set of jobs whose processing times have their membership functions, a fuzzy dispatching algorithm constructs a schedule by means of a final completion time membership function. First, the related fuzzy set operations are briefly reviewed. Then, the fuzzy dispatching algorithms are discussed.

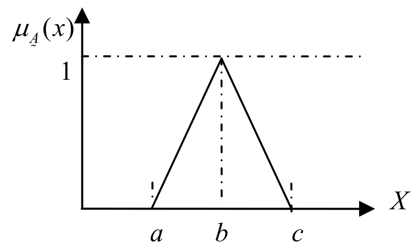


Figure 1. A triangular fuzzy membership function for the fuzzy set $\underset{\sim}{A}$

### 3.1 Related fuzzy set operations

Define two fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ on the universe X. A given element $x$ of the universe is mapped to a *membership value* using a function-theoretic form. Such a function maps elements of a fuzzy set to a real-numbered value from the interval

[0,1]. When the universe of the fuzzy set $\underset{\sim}{A}$ is continuous and infinite, the fuzzy set $\underset{\sim}{A}$ is denoted by (see [8])

$$\underset{\sim}{A} = \left\{ \int \frac{\mu_{\underset{\sim}{A}}(x)}{x} \right\} \tag{1}$$

One type of the function-theoretic forms used in this paper is a triangular membership function. It can be described by $\underset{\sim}{A} = (a, b, c)$, where $a \le b \le c$ (see Figure 1). Its function is as follows:

$$\mu_{\underset{\sim}{A}}(x) = \begin{cases} \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0 & otherwise \end{cases} \tag{2}$$

For a triangular fuzzy membership function, the fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ can be represented as follows:

$$\underset{\sim}{A} = (a_A, b_A, c_A) \text{ and } \underset{\sim}{B} = (a_B, b_B, c_B) \tag{3}$$

The *sum* of the fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$ is obtained as follows:

$$\underset{\sim}{A} + \underset{\sim}{B} = (a_A+a_B, b_A+b_B, c_A+c_B) \tag{4}$$

Many fuzzy ranking methods have been proposed for solving decision and optimization problems such that a good solution can be obtained (see e.g. [9] for a survey). A ranking using the averaging method is one of the most widely used methods [10] and is adopted in this study. The ranking function $R(\underset{\sim}{A})$ is defined as follows:

$$\begin{aligned} R(\underset{\sim}{A}) &= \frac{\int x\mu_{\underset{\sim}{A}}(x)dx}{\int \mu_{\underset{\sim}{A}}(x)dx} \\ &= \frac{\left|\int_a^b (\frac{x-a}{b-a})xdx + \int_b^c (\frac{c-x}{c-b})xdx\right|}{\left|\int_a^b (\frac{x-a}{b-a})dx + \int_b^c (\frac{c-x}{c-b})dx\right|} \\ &= \frac{1}{3}(a+b+c) \end{aligned} \tag{5}$$

Consequently, we say that $\underset{\sim}{A} > \underset{\sim}{B}$ if $R(\underset{\sim}{A}) > R(\underset{\sim}{B})$. In this paper, the operations presented above are used to schedule the jobs with uncertain standard processing times. A triangular membership function is applied to represent the fuzzy standard processing times of each job. It can be denoted by $\underset{\sim}{ps}^t_j = (a_{\underset{\sim}{ps}^t_j}, b_{\underset{\sim}{ps}^t_j}, c_{\underset{\sim}{ps}^t_j})$, where $a_{\underset{\sim}{ps}^t_j} \le b_{\underset{\sim}{ps}^t_j} \le c_{\underset{\sim}{ps}^t_j}$. The average value of the fuzzy standard processing times is represented by $\underset{\sim}{ps}^{t\ ave}_j$.

### 3.2 Heuristic constructions

The fuzzy algorithm for the flexible flow shop problem with unrelated parallel machines is based on dispatching rules (e.g. LPT, SPT and ERD) (see [1-3]), and it uses fuzzy concepts to manage uncertainty. The standard processing times for each job are defined by a fuzzy set. The proposed algorithm is as follows:

**Part 1: Finding the representatives:**

***Step 1***: Select the representatives of the speeds ( $v_{ij}^{/t}$ ) and setup times ( $s_{lj}^{/t}$ ) for every job and every stage by using the combinations of the minimum, maximum, and average data values.

**Part 2: Finding the job sequence:**

***Step 2***: For each job, find the representatives of the fuzzy operating times ( $\underset{\sim}{t}_j^{/t}$ ) and the total fuzzy operating times ( $\underset{\sim}{T}_j^{/}$ ) based on the triangular fuzzy addition operation by using the following equations:

$$\underset{\sim}{t}_j^{/t} = \frac{\underset{\sim}{ps}_j^t}{v_{ij}^{/t}} + s_{lj}^{/t} , \forall t \tag{6}$$

and

$$\underset{\sim}{T}_j^{/} = \sum_{t \in O} \underset{\sim}{t}_j^{/t} \tag{7}$$

where $\underset{\sim}{T}_j^{/} = (a_{\underset{\sim}{T}_j^{/}}, b_{\underset{\sim}{T}_j^{/}}, c_{\underset{\sim}{T}_j^{/}})$ and $a_{\underset{\sim}{T}_j^{/}} \le b_{\underset{\sim}{T}_j^{/}} \le c_{\underset{\sim}{T}_j^{/}}$ .

***Step 3***: For each job, find the average value of the representatives of the fuzzy operating times ( $\underset{\sim}{t}_j^{/t\ ave}$ ) of each stage and the total fuzzy operating times ( $\underset{\sim}{T}_j^{/ave}$ ) by using the following equations:

$$\underset{\sim}{t}_j^{/t\ ave} = \frac{1}{3}(a_{\underset{\sim}{t}_j^{/t}} + b_{\underset{\sim}{t}_j^{/t}} + c_{\underset{\sim}{t}_j^{/t}}) , \ \forall t \tag{8}$$

and

$$\underset{\sim}{T}_j^{/ave} = \frac{1}{3}(a_{\underset{\sim}{T}_j^{/}} + b_{\underset{\sim}{T}_j^{/}} + c_{\underset{\sim}{T}_j^{/}}) \tag{9}$$

***Step 4***: Use the particular dispatching rules to find the first-stage sequence.

Case 1: Sort the jobs in ascending order of the average values of the representatives of the total fuzzy operating times $\underset{\sim}{T}_j^{/ave}$; if any two jobs have the same $\underset{\sim}{T}_j^{/ave}$ values, sort them in an arbitrary order (This algorithm is denoted by $\text{FSPT}^{\text{T}}$).

Case 2: Sort the jobs in ascending order of the average values of the representatives of the fuzzy operating times $\underset{\sim}{t}_j^{/t\ ave}$ of each stage; if any two jobs have the same $\underset{\sim}{t}_j^{/t\ ave}$ values, sort them in an arbitrary order (These algorithms are denoted by $\text{FSPT}^1$, $\text{FSPT}^2$, …, $\text{FSPT}^{\text{k}}$).

Case 3: Sort the jobs in descending order of the average values of the representatives of the total fuzzy operating times $\underset{\sim}{T}_j^{/ave}$; if any two jobs have the same $\underset{\sim}{T}_j^{/ave}$ values, sort them in an arbitrary order (This algorithm is denoted by $\text{FLPT}^{\text{T}}$).

Case 4: Sort the jobs in descending order of the average values of the representatives of the fuzzy operating times $\underset{\sim}{t}_j^{/t\ ave}$ of each stage; if any two jobs have the same $\underset{\sim}{t}_j^{/t\ ave}$ values, sort them in an arbitrary order (These algorithms are denoted by $\text{FLPT}^1$, $\text{FLPT}^2$, …, $\text{FLPT}^{\text{k}}$).

Case 5: Sort the jobs in ascending order of the release dates $r_j$ of jobs; if any two jobs have the same $r_j$ values, sort them in an arbitrary order (This algorithm is denoted by FERD).

**Part 3: Assigning the jobs to the machines at the first stage**

***Step 5***: Assign the first job $j_{[1]}$ in the ordered job sequence to the machine which has the minimum average fuzzy completion time among all machines of this stage.

***Step 6***: Update the availability ( $\underset{\sim}{a}_i^1$ ) of the selected machine $i$ by using the value of the fuzzy completion time of the job assigned to this machine.

***Step 7***: Remove the job from the ordered job sequence.

***Step 8***: Repeat Steps 5 to 7 until the job sequence is empty.

**Part 4: Assigning the jobs to the machines at the other stages**

***Step 8***: Find the job sequence of the next stage $t$.

Case 1: Set the job sequence for the stage to be equal to the ordered job sequence obtained in Step 4 (permutation rule).

Case 2: Determine the job sequence for the current stage by ordering the jobs according to their average fuzzy completion times at the previous stage (FIFO rule).

***Step 9***: Assign the first job $j_{[1]}$ in the job sequence in Step 8 to the machine which has the minimum average fuzzy completion time among all machines of the stage.

***Step 10***: Update the availability ( $\underset{\sim}{a}_i^t$ ) of the selected machine $i$ by using the value of the average fuzzy completion time of the job assigned to this machine.

***Step 11***: Remove the job from the ordered job sequence.

***Step 12***: Repeat Steps 8 to 11 until the job sequence is empty.

***Step 13***: Consider the next stage and Repeat Steps 8 to 12 until stage $k$ has been considered.

**Part 5: Finding the best solution**

***Step 14***: Repeat Steps 2 to 13 for the other representatives, and return the best fuzzy solution with $\underset{\sim}{C}_{max}$ = ( $a_{\underset{\sim}{C}_{max}}$ , $b_{\underset{\sim}{C}_{max}}$ , $c_{\underset{\sim}{C}_{max}}$ ) and the average value $\underset{\sim}{C}_{max}^{ave}$ .

## 4. A NUMERICAL EXAMPLE

A numerical example is provided in this section to illustrate the algorithm proposed. Let $n = 5$, $k = 2$, and $m^1 = 2$ and $m^2 = 1$. The release dates of jobs are 9, 19, 0, 7 and 0, respectively. The machine availabilities are 36 and 14 for the machines at the first stage and 104 for the machine at the second stage. Moreover, assume the fuzzy standard processing times given in Table 1. The relative speeds of machines are shown in Table 2. The sequence- and machine-dependent setup times are given in Table 3 and Table 4, respectively. The algorithm works as follows.

Table 1. Fuzzy standard processing times

| | $\underset{\sim}{ps}_j^1$ | $\underset{\sim}{ps}_j^2$ |
|---|---|---|
| Job 1 | (76, 85, 95) | (81, 88, 94) |
| Job 2 | (59, 67, 71) | (49, 59, 62) |
| Job 3 | (88, 95, 99) | (31, 33, 41) |
| Job 4 | (69, 78, 84) | (91, 95, 101) |
| Job 5 | (62, 62, 68) | (75, 76, 76) |

Table 2. Relative speeds of machines

| | **Job 1** | **Job 2** | **Job 3** | **Job 4** | **Job 5** |
|---|---|---|---|---|---|
| $v_{1j}^1$ | 1.132 | 1.180 | 0.706 | 1.138 | 0.730 |
| $v_{2j}^1$ | 0.838 | 0.802 | 1.000 | 1.288 | 1.102 |
| $v_{1j}^2$ | 1.138 | 1.168 | 0.946 | 1.174 | 0.946 |

Table 3. Sequence-dependent setup times

| | **Job 1** | **Job 2** | **Job 3** | **Job 4** | **Job 5** |
|---|---|---|---|---|---|
| $s_{1l}^1$ | X | 14 | 19 | 45 | 7 |
| $s_{2l}^1$ | 5 | X | 9 | 12 | 30 |
| $s_{3l}^1$ | 21 | 36 | X | 22 | 27 |
| $s_{4l}^1$ | 19 | 8 | 31 | X | 26 |
| $s_{5l}^1$ | 22 | 23 | 46 | 30 | X |
| $s_{1l}^2$ | X | 45 | 36 | 47 | 7 |
| $s_{2l}^2$ | 13 | X | 31 | 15 | 13 |
| $s_{3l}^2$ | 34 | 5 | X | 11 | 20 |
| $s_{4l}^2$ | 4 | 50 | 32 | X | 26 |
| $s_{5l}^2$ | 15 | 11 | 44 | 34 | X |

Table 4. Machine-dependent setup times

| | Job 1 | Job 2 | Job 3 | Job 4 | Job 5 |
|---|---|---|---|---|---|
| $ch_{1j}^{1}$ | 23 | 37 | 18 | 30 | 30 |
| $ch_{2j}^{1}$ | 10 | 36 | 32 | 36 | 43 |
| $ch_{1j}^{2}$ | 23 | 4 | 43 | 15 | 12 |

Table 5. Representatives of the fuzzy operating times (by using $\text{setup}_{\min}$ and $\text{speed}_{\min}$)

| | $\underset{\sim}{t}_j^{/1}$ | $\underset{\sim}{t}_j^{/2}$ |
|---|---|---|
| Job 1 | (95.692, 106.432, 118.365) | (75.178, 81.329, 86.601) |
| Job 2 | (81.566, 91.541, 96.529) | (45.952, 54.514, 57.082) |
| Job 3 | (133.646, 143.561, 149.227) | (63.770, 65.884, 74.340) |
| Job 4 | (72.633, 80.541, 85.814) | (88.513, 91.920, 97.031) |
| Job 5 | (91.932, 91.932, 100.151) | (86.281, 87.338, 87.338) |

Table 6. Representatives of the total fuzzy operating times

| | $\underset{\sim}{T}_j^{/}$ | $\underset{\sim}{T}_j^{/ave}$ |
|---|---|---|
| Job 1 | (170.870, 187.761, 204.966) | 187.866 |
| Job 2 | (127.518, 146.055, 153.611) | 142.395 |
| Job 3 | (197.416, 209.445, 223.567) | 210.143 |
| Job 4 | (161.146, 172.461, 182.845) | 172.151 |
| Job 5 | (178.213, 179.270, 187.489) | 181.657 |

**Part 1: Finding the representatives:**

***Step 1***: Select the minimum values of the speeds and the setup times for every job and every stage.

**Part 2: Finding the job sequence:**

***Step 2***: For each job, find the representatives of the fuzzy operating times and the total fuzzy operating times based on the triangular fuzzy addition operation. The results are shown in Table 5 and Table 6, respectively.

***Step 3***: Assume that we use the $\text{FSPT}^{T}$ algorithm. Find the average values of the representatives of the total fuzzy operating times, the results are shown in the last column of Table 6.

***Step 4***: Sort the jobs in ascending order of the average values of representatives of the total fuzzy operating times $\underset{\sim}{T}_j^{/ave}$. Thus, the ordered job sequence is $\{2, 4, 5, 1, 3\}$.

**Part 3: Assigning the jobs to the machines at the first stage**

***Step 5***: Assign the first job $j_{[1]}$ (job 2) in the ordered job sequence to the machine which has the minimum average fuzzy completion time. The results are as follows.

-on machine 1: $\max\{\underset{\sim}{a}_1^{1ave}, \underset{\sim}{r}_2^{ave}\} + ch_{12}^{1} + \dfrac{\underset{\sim}{ps}_2^{1\,ave}}{v_{12}^{1}} = 128.650;$

-on machine 2: $\max\{\underset{\sim}{a}_2^{1ave}, \underset{\sim}{r}_2^{ave}\} + ch_{22}^{1} + \dfrac{\underset{\sim}{ps}_2^{1\,ave}}{v_{22}^{1}} = 136.879;$

Thus, job 2 is assigned to machine 1, since the average fuzzy completion time of job 2 assigned to this machine is lower than the average fuzzy completion time on the other machine. Its fuzzy completion time is (123.000, 129.780, 133.170).

***Step 6***: Set the availability of machine 1 to be (123.000, 129.780, 133.170). In addition, we set the release date of job 2 for the next stage to be (123.000, 129.780, 133.170) as well.

***Step 7***: Remove job 2 from the ordered job sequence.

***Step 8***: Repeat Steps 5 to 7 until the job sequence is empty. For the next job (i.e. job 4), the average fuzzy completion time is calculated as follows.

-on machine 1: $\max\{\underset{\sim}{a}_1^{1ave}, \underset{\sim}{r}_4^{ave}\} + s_{24}^{1} + \dfrac{\underset{\sim}{ps}_4^{1\,ave}}{v_{14}^{1}} = 208.312;$

-on machine 2: $\max\{\underset{\sim}{a}_2^{1ave}, \underset{\sim}{r}_4^{ave}\} + ch_{24}^{1} + \dfrac{\underset{\sim}{ps}_4^{1\,ave}}{v_{24}^{1}} = 109.783;$

Again, job 4 is assigned to machine 2, since the average fuzzy completion time is shorter in this case. All results are shown in Table7.

Table 7. Fuzzy completion times at stage 1

| Job # | Machine # | Fuzzy completion times | Average fuzzy completion times |
|---|---|---|---|
| 2 | 1 | (123.000, 129.780, 133.169) | 128.650 |
| 4 | 2 | (103.571, 110.559, 115.217) | 109.783 |
| 5 | 2 | (185.833, 192.820, 202.923) | 193.859 |
| 1 | 1 | (195.138, 209.868, 222.092) | 209.033 |
| 3 | 2 | (319.833, 333.820, 347.923) | 333.859 |

Table 8. Fuzzy completion times at stage 2 (by using $setup_{min}$ and $speed_{min}$)

| Job # | Machine # | Fuzzy completion times | Average fuzzy completion times |
|---|---|---|---|
| *Case 1: Permutation rule* | | | |
| 2 | 1 | (168.952, 184.293, 190.252) | 181.166 |
| 4 | 1 | (261.465, 280.213, 291.282) | 277.653 |
| 5 | 1 | (366.746, 386.552, 397.621) | 383.639 |
| 1 | 1 | (452.924, 478.880, 495.222) | 475.675 |
| 3 | 1 | (521.693, 549.764, 574.562) | 548.673 |
| *Case 2: FIFO rule* | | | |
| 4 | 1 | (196.513, 206.479, 216.248) | 206.413 |
| 2 | 1 | (288.465, 306.993, 319.330) | 304.929 |
| 5 | 1 | (380.746, 400.331, 412.669) | 397.915 |
| 1 | 1 | (466.924, 492.660, 510.270) | 489.951 |
| 3 | 1 | (535.693, 563.543, 589.610) | 562.949 |

**Part 4: Assigning the jobs to the machines at the other stages**

***Step 8***: Find the job sequence for the next stage $t$.

Case 1: For the permutation rule, set the job sequence to be equal to {2, 4, 5, 1, 3}.

Case 2: For the FIFO rule, set the job sequence to be equal to {4, 2, 5, 1, 3}.

***Step 9***: Again as in Step 5, assign the first job $j_{[1]}$ (job 2 in case 1 and job 4 in case 2) to the machine which has the minimum average fuzzy completion time. However, in this example, there is only one machine at the second stage, so in case 1 (the permutation rule) job 2 is assigned to the machine first, whereas job 4 is assigned to the machine first otherwise (the FIFO rule).

***Step 10***: Update the availability of the machine by using the value of the average fuzzy completion time of the job assigned to this machine.

***Step 11***: Remove job 2 for the permutation rule (and job 4 for the FIFO rule) from the ordered job sequence.

***Step 12***: Repeat Steps 8 to 11 until the job sequence is empty.

***Step 13***: Consider the next stage and Repeat Steps 8 to 12 until stage $k$ has been considered (but for this example, we have $k$ =2). The results are shown in Table 8.

For the chosen representatives of the fuzzy operating times, the permutation rule generates a solution which is better than that generated by the FIFO rule. Hence, we select the solution generated by the permutation rule and the fuzzy completion time is (521.693, 549.764, 574.562).

**Part 5: Finding the best solution**

***Step 14***: Repeat Steps 2 to 13 for the other representatives, and return the best fuzzy solution. The results for this example are shown in Table 9.

Table 9. The best fuzzy solution

| Job # | Machine # | Fuzzy completion times | Average fuzzy completion times |
|---|---|---|---|
| *Stage1:* | | | |
| 2 | 1 | (123.000, 129.780, 133.169) | 128.650 |
| 5 | 2 | (113.261, 113.261, 118.706) | 115.076 |
| 4 | 2 | (196.833, 203.820, 213.923) | 204.859 |
| 1 | 1 | (195.138, 209.868, 222.092) | 209.033 |
| 3 | 2 | (315.833, 329.820, 343.923) | 329.859 |
| *Stage2:* | | | |
| 5 | 1 | (204.543, 205.600, 211.044) | 207.062 |
| 2 | 1 | (257.495, 267.113, 275.126) | 266.578 |
| 4 | 1 | (350.007, 363.033, 376.157) | 363.066 |
| 1 | 1 | (425.185, 444.362, 462.758) | 444.102 |
| 3 | 1 | (493.954, 515.246, 542.099) | 517.100 |

For the FLPT$^T$ and FERD algorithms, the results are (505.693, 533.543, 559.610) and (582.693, 610.984, 636.393), respectively. For determining an optimal solution, we have used the standard processing times by using the values $a_{\tilde{ps}_j^t}$, $b_{\tilde{ps}_j^t}$, $c_{\tilde{ps}_j^t}$, and $\tilde{ps}_j^{t\ ave}$ as the standard processing times in the mathematical model (see Jungwattanakit, Reodecha, Chaovalitwongse, and Werner [1]). The results of the completion times are shown in Table 10.

Table 10. An optimal solution

| **Case of $\tilde{ps}_j^t$** | **Completion times** |
|---|---|
| $a_{\tilde{ps}_j^t}$ | 474.693 |
| $b_{\tilde{ps}_j^t}$ | 512.543 |
| $c_{\tilde{ps}_j^t}$ | 528.609 |
| $\tilde{ps}_j^{t\ ave}$ | 501.807 |

## 5. COMPUTATIONAL RESULTS

In our tests, we used problems with 5, 10, 20, and 100 jobs, 2 and 5 machines per stage, and 2 and 10 stages. For each problem size, ten different instances have been run. The standard processing times are fuzzy such that the values of $b_{\tilde{ps}_j^t}$ are generated uniformly from the interval [10,100], $a_{\tilde{ps}_j^t} = b_{\tilde{ps}_j^t}$ - 10×U[0,1] and $c_{\tilde{ps}_j^t} = b_{\tilde{ps}_j^t}$ + 10×U[0,1], where U[0,1] denotes a uniformly distributed random number from the interval [0,1]. The relative speeds are distributed uniformly in the interval [0.7, 1.3]. The setup times, both sequence- and machine-dependent setup times, are uniformly generated from the interval [0, 50], whereas the release dates are uniformly generated from the interval between 0 and half of their total standard processing time mean.

First, we present the results of the fuzzy dispatching algorithms for small-size two-stage problems with five jobs and two machines per stage. We give the average deviation from the optimal makespan value obtained by using a commercial mathematical programming software, CPLEX 8.0.0 and AMPL, with an Intel Pentium 4 2.00GHz CPU with 256 MB of RAM.

For the small-size problems, we obtained the average optimal values and the average makespan values of the fuzzy dispatching algorithms as shown in Table 11. The average optimal solutions using the fuzzy standard processing time values $a_{\tilde{ps}_j^t}$, $b_{\tilde{ps}_j^t}$, $c_{\tilde{ps}_j^t}$, and $\tilde{ps}_j^{t\ ave}$ are 268.589, 283.727, 301.787, and 283.997, respectively. It can be observed that for the fuzzy SPT algorithms, an FSPT$^1$ algorithm outperforms the others, whereas an FSPT$^2$ algorithm gives poor solutions. For the fuzzy LPT algorithms, an FLPT$^2$ algorithm significantly outperforms the others, whereas an FLPT$^1$ algorithm gives poor solutions. The results show that the average percentage deviations from the average optimal fuzzy makespan values of an FLPT$^2$ algorithm are about 5 percent.

Table 11. Average performance of the fuzzy dispatching algorithms for small-size test problems

| **Algorithms** | | $\tilde{C}_{max}$ | **%deviation** | **Algorithms** | | $\tilde{C}_{max}$ | **%deviation** | **Algorithms** | | $\tilde{C}_{max}$ | **%deviation** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FSPT$^T$ | *a* | 330.843 | 23.178 | FLPT$^T$ | *a* | 306.480 | 14.108 | FERD | *a* | 314.400 | 17.056 |
| | *b* | 348.398 | 22.793 | | *b* | 327.611 | 15.467 | | *b* | 329.088 | 15.988 |
| | *c* | 369.814 | 22.541 | | *c* | 346.916 | 14.954 | | *c* | 348.205 | 15.381 |
| | *ave* | 349.685 | 23.130 | | *ave* | 327.002 | 15.143 | | *ave* | 330.564 | 16.397 |
| FSPT$^1$ | *a* | 313.889 | 16.866 | FLPT$^1$ | *a* | 328.988 | 22.488 | | | | |
| | *b* | 328.482 | 15.774 | | *b* | 348.015 | 22.659 | | | | |
| | *c* | 345.702 | 14.552 | | *c* | 367.626 | 21.816 | | | | |
| | *ave* | 329.358 | 15.972 | | *ave* | 348.210 | 22.610 | | | | |
| FSPT$^2$ | *a* | 347.589 | 29.413 | FLPT$^2$ | *a* | 282.449 | 5.160 | | | | |
| | *b* | 364.628 | 28.514 | | *b* | 297.872 | 4.986 | | | | |
| | *c* | 382.702 | 26.812 | | *c* | 315.784 | 4.638 | | | | |
| | *ave* | 364.973 | 28.513 | | *ave* | 298.701 | 5.178 | | | | |

Next, for large-size problems, we cannot find an optimal solution within a reasonable time since the problem under consideration is NP-hard. Thus, we used the fuzzy dispatching algorithms to find a best solution instead of an optimal solution. The results in Table 12 show the average values of the average fuzzy completion times ( $\underset{\sim}{C}_{max}^{ave}$ ) of ten different instances for each problem size. It can be seen that among the fuzzy SPT algorithms, the $FSPT^1$ algorithm sorting the jobs in ascending order of the average values of the fuzzy operating times $\underset{\sim}{t}_j^{/1}$ gives the best solutions, whereas among the fuzzy LPT algorithms, the $FLPT^k$ algorithm sorting the jobs in descending order of the average values of the fuzzy operating times $\underset{\sim}{t}_j^{/k}$ gives the best solutions. Although the $FSPT^2$ algorithm outperforms the $FSPT^1$ algorithm for the test problems with 5 jobs, 2 machines per stage and 10 stages and 20 jobs, 5 machines per stage and 10 stages, its value is slightly worse than the value of the $FSPT^1$ algorithm. In addition, the $FLPT^5$ algorithm outperforms the other fuzzy LPT algorithms for the test problems with 5 jobs, 2 machines per stage, and 10 stages but for the other problems, the $FLPT^k$ algorithm is clearly the best. In general, the quality of an $FSPT^t$ algorithm improves with a decreasing value of *t*, whereas the quality of an $FLPT^t$ algorithm improves with an increasing value of *t*. Comparing all fuzzy dispatching algorithms, we have found that we can recommend the $FLPT^k$ algorithm for the problem under consideration. Moreover, we have found that we can improve the quality of the SPT and LPT algorithms, which have been applied to deterministic processing times in [1-3], by using the operating times at stage 1 and stage *k*, respectively, to sequence the jobs instead of the total operating times.

Table 12. Average values of the average fuzzy completion times ( $\underset{\sim}{C}_{max}^{ave}$ ) for the problem tests

| Problems (n/m/t)* | $FSPT^T$ | $FSPT^1$ | $FSPT^2$ | $FSPT^3$ | $FSPT^4$ | $FSPT^5$ | $FSPT^6$ | $FSPT^7$ | $FSPT^8$ | $FSPT^9$ | $FSPT^{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (5/2/2) | 349.685 | **329.358** | 364.973 | | | | | | | | |
| (5/2/10) | 1016.979 | 1023.216 | **1014.870** | 1022.886 | 1035.311 | 1021.089 | 1030.636 | 1031.053 | 1024.931 | 1031.238 | 1044.929 |
| (10/2/2) | 526.921 | **498.975** | 533.673 | | | | | | | | |
| (10/2/10) | 1304.503 | **1268.252** | 1294.020 | 1315.095 | 1303.967 | 1299.899 | 1286.042 | 1356.330 | 1330.665 | 1314.459 | 1321.216 |
| (10/5/2) | 291.475 | **275.391** | 293.138 | | | | | | | | |
| (10/5/10) | 942.851 | **918.613** | 924.823 | 933.835 | 939.381 | 926.116 | 934.014 | 943.274 | 945.333 | 944.080 | 954.470 |
| (20/2/2) | 977.859 | **914.338** | 1033.777 | | | | | | | | |
| (20/2/10) | 1784.016 | **1710.099** | 1740.533 | 1760.855 | 1793.445 | 1772.116 | 1819.504 | 1789.933 | 1816.277 | 1839.396 | 1841.654 |
| (20/5/2) | 443.928 | **428.400** | 457.022 | | | | | | | | |
| (20/5/10) | 1095.988 | 1066.886 | **1064.110** | 1085.275 | 1088.496 | 1075.059 | 1091.359 | 1084.104 | 1091.938 | 1091.091 | 1101.908 |
| (100/2/2) | 4103.486 | **3972.679** | 4534.109 | | | | | | | | |
| (100/2/10) | 5353.308 | **5244.261** | 5344.762 | 5426.015 | 5433.469 | 5554.546 | 5590.312 | 5695.343 | 5720.433 | 5719.129 | 5767.628 |
| (100/5/2) | 1592.607 | **1534.279** | 1758.996 | | | | | | | | |
| (100/5/10) | 2427.176 | **2330.101** | 2390.710 | 2460.560 | 2434.737 | 2483.375 | 2500.791 | 2535.864 | 2534.658 | 2540.412 | 2550.593 |

| Problems (n/m/t) | $FLPT^T$ | $FLPT^1$ | $FLPT^2$ | $FLPT^3$ | $FLPT^4$ | $FLPT^5$ | $FLPT^6$ | $FLPT^7$ | $FLPT^8$ | $FLPT^9$ | $FLPT^{10}$ | FERD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| (5/2/2) | 327.002 | 348.210 | **298.701** | | | | | | | | | 330.564 |
| (5/2/10) | 1017.347 | 1033.726 | 1041.223 | 1045.443 | 1029.815 | **1003.797** | 1035.721 | 1017.153 | 1015.295 | 1020.752 | 1010.479 | 1024.262 |
| (10/2/2) | 491.604 | 520.869 | **478.012** | | | | | | | | | 532.770 |
| (10/2/10) | 1311.684 | 1344.721 | 1333.110 | 1336.311 | 1299.907 | 1328.251 | 1328.775 | 1316.256 | 1282.278 | 1287.543 | **1276.509** | 1346.127 |
| (10/5/2) | 245.055 | 269.105 | **240.404** | | | | | | | | | 282.218 |
| (10/5/10) | 928.938 | 937.630 | 934.272 | 937.766 | 934.940 | 932.524 | 943.040 | 938.085 | 921.840 | 933.132 | **918.378** | 945.896 |
| (20/2/2) | 933.492 | 1022.701 | **892.706** | | | | | | | | | 979.634 |
| (20/2/10) | 1789.709 | 1852.273 | 1821.411 | 1829.180 | 1798.512 | 1796.631 | 1781.070 | 1774.708 | 1757.243 | 1740.589 | **1719.326** | 1794.556 |
| (20/5/2) | 403.685 | 454.101 | **387.865** | | | | | | | | | 453.302 |
| (20/5/10) | 1070.125 | 1113.055 | 1094.801 | 1094.112 | 1078.427 | 1083.758 | 1076.538 | 1078.216 | 1072.151 | 1074.903 | **1055.036** | 1092.426 |
| (100/2/2) | 4042.877 | 4488.094 | **3951.540** | | | | | | | | | 4150.841 |
| (100/2/10) | 5347.711 | 5749.591 | 5770.639 | 5668.099 | 5597.765 | 5583.685 | 5565.207 | 5462.565 | 5432.274 | 5340.251 | **5280.089** | 5399.334 |
| (100/5/2) | 1555.362 | 1743.006 | **1511.234** | | | | | | | | | 1617.421 |
| (100/5/10) | 2409.504 | 2556.120 | 2542.728 | 2508.510 | 2490.274 | 2491.579 | 2466.467 | 2442.547 | 2429.643 | 2409.451 | **2344.116** | 2428.159 |

*(n/m/t) = (number of jobs/ number of machines per stage/ number of stages)

## 6. CONCLUSIONS

In this paper, fuzzy dispatching rules (namely LPT, SPT and ERD) have been investigated for minimizing the makespan for the flexible flow shop problem with unrelated parallel machines and setup times, which is often occurring in real world problems. Such algorithms are based on the list scheduling principle by developing job sequences for the first stage and assigning and sequencing the remaining stages by both the permutation and FIFO approaches. In addition, processing times under uncertainty have been considered. We have solved this problem by using fuzzy set theory. In particular, we used a triangular membership function for the standard processing times to get a more real-world application. Thus, fuzzy dispatching rules are proposed to manage jobs with uncertain standard processing times. This approach generates a scheduling result with a membership function completion time. Among the fuzzy SPT algorithms, the $FSPT^1$ algorithm gives the best solutions, whereas among the fuzzy LPT algorithms, the $FLPT^k$ algorithm gives the best solutions. In general, the $FLPT^k$ algorithm that uses the average values of the fuzzy operating times of the last stage gives the best solutions for both small- and large-size test problems among all fuzzy dispatching algorithms considered in this paper. In particular, the results show that the recommended fuzzy LPT algorithm gives a deviation from the optimal makespan value of about five percent for small-size test problems. Finally, the better the estimate we can have for the processing times, the less uncertainty we get for the fuzzy makespan.

In the future, we will use other algorithms for this problem and try to apply other characteristics of fuzzy sets to the scheduling area. For instance, we can apply other types of membership functions to this or other scheduling problems.

## ACKNOWLEDGMENT

This work was supported in part by INTAS (project 03-51-5501) and the Department of Industrial Engineering, Chulalongkorn University.

## REFERENCES

1. Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P. and Werner, F. (2007). Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. The International Journal of Advanced Manufacturing Technology. (in press) DOI 10.1007/s00170-007-0977-0.

2. Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P. and Werner, F. (2007). Constructive and tabu search algorithms for hybrid flow shop problems with unrelated parallel machines and setup times. International Journal of Computational Science. 1(2): 204-214.

3. Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P. and Werner, F. (2007). Constructive and simulated annealing algorithms for hybrid flow shop problems with unrelated parallel machines. Thammasat International Journal of Science and Technology. 12(1): 31-41.

4. Balasubramanian, J. and Grossmann, I.E. (2003). Scheduling optimization under uncertainty – and alternative approach. Computers & Chemical Engineering. 27(4): 469-490.

5. Han, S., Ishii, H. and Fujii, S. (1994). One machine scheduling problem with fuzzy duedates. European Journal of Operational Research. 79(1): 1-12.

6. Gen, M., Tsujimura, Y. and Zheng, D. (1997). An application of fuzzy set theory to inventory control models. Computers & Industrial Engineering. 33(3-4): 553-556.

7. Kuroda, M. and Wang, Z. (1996). Fuzzy job shop scheduling. International Journal of Production Economics. 44(1-2): 45-51.

8. Ross, T.J. (1995). Fuzzy logic with engineering applications. McGraw-Hill, New York, U.S.

9. Chang, P.T. and Lee, E.S. (1994). Ranking of fuzzy sets based on the concept of existence. Computer & Mathematics with Applications. 27(9-10): 1-21.

10. Sun, K.T. (1998). Job scheduling using ranking fuzzy number method. Proceedings of 1998 2nd International conference on Knowledge-Based Intelligent Electronic Systems, Adelaide, SA, Australia, Apr. 21-23, pp. 97-101.

APIEMS & CIIE 2007
KAOHSIUNG

***Sponsor:***

Asia Pacific Industrial Engineering and Management Society
Chinese Institute of Industrial Engineers

***Conference Organizer:***

Department of Industrial Engineering and Management
National Yunlin University of Science and Technology

***Advisory organization:***

National Science Council
Ministry of Economic Affairs, R.O.C.
Ministry of Education, R.O.C.
The Government of Kaohsiung City