คณะวิศวกรรมศาสตร์
มหาวิทยาลัยสงขลานครินทร์
เลิศล้ำวิชาการ สรรค์สร้างเทคโนโลยี 40 ปี วิศวฯ มอ.

รวมบทคัดย่อ

The mermaid at Samila beach, Songkhla.

IE Network Conference 2007

24-26 October 2007
Royal Phuket City Hotel, Phuket

The IE Network Conference
24-26 October 2007

Fuzzy LPT Algorithms for Flexible Flow Shop Problems with Unrelated Parallel Machines for a Continuous Fuzzy Domain

Jitti Jungwattanakit[1*] Manop Reodecha[2] Paveena Chaovalitwongse[3] Frank Werner[4]
[1,2,3] Department of Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330
[4]Faculty of Mathematics, Otto-von-Guericke-University, P.O. Box 4120, D-39016 Magdeburg, Germany
E-mail: jitti.j@student.chula.ac.th[*]

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 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 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 human factors or operating faults. In this paper, fuzzy concepts are used in an LPT algorithm for managing uncertain scheduling. Given a set of jobs together with a membership function for the standard processing times, the fuzzy LPT algorithms construct a solution by means of a membership function for the final makespan. The proposed algorithms provide a more flexible method of scheduling jobs than conventional scheduling methods. The results show that the fuzzy LPT algorithms give a deviation from the optimal makespan value of about five percent for the small-size test problems. In addition, the fuzzy LPT algorithm by using the average values of the total operating times at the last stage (denoted by FLPT^k) gives a good solution for both small- and large-test problems.

Keywords: Flexible Flow Shop, Unrelated Parallel Machines, Fuzzy Sets, LPT Algorithm

1. Introduction

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 a number of stages in the same order. Moreover, 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. 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 LPT algorithm is a good dispatching rule 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 human factors or operating faults. The estimated processing times are not precisely known. Consequently, several concepts such as fuzzy set theory, probability theory, DEMPSTER/SHAFER theory, a sensitivity analysis, and others, have been used to take into account the uncertainties.

In this paper, we will treat uncertainty by using fuzzy set theory because of its simplicity and similarity to human reasoning [4]. Such a theory has been applied to many areas such as inventory control [5] and scheduling [6]. We apply fuzzy LPT 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 LPT algorithms. A numerical example is discussed in Section 4. Computational results are discussed in Section 5 and conclusions are given in Section 6.

2. Problem Description

Flexible flow shop problems can be described as follows. There is a set $J = \{1,\ldots, j,\ldots, n\}$ of n independent jobs which need to be processed, and the processing system is defined by a set $O = \{1,\ldots, t,\ldots, k\}$ of k processing stages. At each stage t, $t \in O$, there is a set $M^t = \{1,\ldots, i,\ldots, m^t\}$ 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 at 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 $\tilde{ps}_j^t$ 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. 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 $\tilde{C}_{max}$ which is equivalent to the fuzzy completion time of the last job leaving the system.

3. Fuzzy LPT Scheduling Algorithms

In this section, fuzzy set theory is used in an LPT scheduling algorithm to schedule the jobs with uncertain standard processing times. Given a set of jobs whose processing times have their membership functions, the fuzzy LPT algorithms construct a schedule by means of a final completion time membership function. First, the related fuzzy set operations are briefly reviewed. Then, the fuzzy LPT algorithms are proposed.

3.1 Related fuzzy set operations

Define two fuzzy sets $\tilde{A}$ and $\tilde{B}$ 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 $\tilde{A}$ is continuous and infinite, the fuzzy set $\tilde{A}$ is denoted by (see [7])

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

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

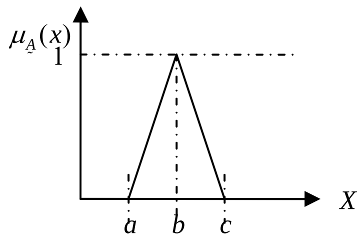

Figure 1. A triangular fuzzy membership function for the fuzzy set $\tilde{A}$.

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

$$\tilde{A} = (a_A, b_A, c_A) \text{ and } \tilde{B} = (a_B, b_B, c_B) \tag{2}$$

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

$$\tilde{A} + \tilde{B} = (a_A + a_B, b_A + b_B, c_A + c_B) \tag{3}$$

Many fuzzy ranking methods have been proposed for solving decision and optimization problems such that a good solution can be obtained (see e.g. [8] for a survey). A ranking using the averaging method is one of the most widely used methods [9] and is adopted in this study. The ranking function $R(\tilde{A})$ is defined as follows:

$$\begin{aligned} R(\tilde{A}) &= \frac{\int x \mu_{\tilde{A}}(x)dx}{\int \mu_{\tilde{A}}(x)dx} \\ &= \frac{\left| \int_a^b (\frac{x-a}{b-a}) x dx + \int_b^c (\frac{c-x}{c-b}) x dx \right|}{\left| \int_a^b (\frac{x-a}{b-a}) dx + \int_b^c (\frac{c-x}{c-b}) dx \right|} \\ &= \frac{1}{3}(a+b+c) \end{aligned} \tag{4}$$

Therefore, we say that $\tilde{A} > \tilde{B}$ if $R(\tilde{A}) > R(\tilde{B})$. 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 $\tilde{ps}_j^t = (a_{\tilde{ps}_j^t}, b_{\tilde{ps}_j^t}, c_{\tilde{ps}_j^t})$, where $a_{\tilde{ps}_j^t} \leq b_{\tilde{ps}_j^t} \leq c_{\tilde{ps}_j^t}$. The average value of the fuzzy standard processing times is represented by $\tilde{ps}_j^{t\ ave}$.

3.2 Heuristic constructions

The fuzzy algorithm for the flexible flow shop problem with unrelated parallel machines is based on the LPT rule, 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 $s_{lj}^{/t}$ and setup times $v_{ij}^{/t}$ 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 ($\tilde{t}_j^{/t}$) and the total fuzzy operating times ($\tilde{T}_j^{/}$) based on the triangular fuzzy addition operation by using the following equations:

$$\tilde{t}_j^{/t} = \frac{\tilde{ps}_j^t}{v_{ij}^{/t}} + s_{lj}^{/t}, \ \forall t \tag{5}$$

and

$$\tilde{T}_j^{/} = \sum_{t \in O} \tilde{t}_j^{/t} \tag{6}$$

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

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

$$\tilde{t}_j^{/t\ ave} = \frac{1}{3}\,(a_{\tilde{t}_j^{/t}} + b_{\tilde{t}_j^{/t}} + c_{\tilde{t}_j^{/t}}), \ \forall t \tag{7}$$

and

$$\tilde{T}_j^{/ave} = \frac{1}{3}\,(a_{\tilde{T}_j^{/}} + b_{\tilde{T}_j^{/}} + c_{\tilde{T}_j^{/}}) \tag{8}$$

Step 4: Use the fuzzy LPT algorithms to find the first-stage sequence.

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

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

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 ($\tilde{a}_i^1$) of the selected machine 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.

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 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 ($\tilde{a}_i^t$) of the selected machine by using the value of the average fuzzy completion time of the job assigned to the 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: Return the best fuzzy solution with $\tilde{C}_{max}$ = ($a_{\tilde{C}_{max}}$, $b_{\tilde{C}_{max}}$, $c_{\tilde{C}_{max}}$) and the average value $\tilde{C}_{max}^{ave}$.

Table 1 Fuzzy standard processing times.

	$\tilde{ps}_j^1$	$\tilde{ps}_j^2$
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 the 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

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 the 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:

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 FLPT^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 descending order of the average values of representatives of the total fuzzy operating times. Thus, the ordered job sequence is {3, 1, 5, 4, 2}.

Table 5 Representatives of the fuzzy operating times (by using setup$_{min}$ and 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 3: Assigning the jobs to the machines at the first stage
Step 5: Assign the first job $j_{[1]}$ (job 3) in the ordered job sequence to the machine which has the minimum average fuzzy completion time. The results are as follows:

-on machine 1: $\max\{\underset{\sim}{a}_1^{1ave}, \underset{\sim}{r}_3^{ave}\} + ch_{13}^1 + \frac{\underset{\sim}{ps}_3^{1\,ave}}{v_{13}^1}$ = 187.145;

-on machine 2: $\max\{\underset{\sim}{a}_2^{1ave}, \underset{\sim}{r}_3^{ave}\} + ch_{23}^1 + \frac{\underset{\sim}{ps}_3^{1\,ave}}{v_{23}^1}$ = 140.000;

Thus, job 3 is assigned to machine 2, since the average fuzzy completion time of job 3 assigned to this machine is lower than the average fuzzy completion time on the other machine. Its fuzzy completion time is (134, 141, 145).
Step 6: Set the availability of machine 2 to be (134, 141, 145). In addition, we set the release date of job 3 for the next stage to be (134, 141, 145) as well.
Step 7: Remove job 3 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 1), the average fuzzy completion time is calculated as follows.

-on machine 1: $\max\{\underset{\sim}{a}_1^{1ave}, \underset{\sim}{r}_1^{ave}\} + ch_{11}^1 + \frac{\underset{\sim}{ps}_1^{1\,ave}}{v_{11}^1}$ = 134.383;

-on machine 2: $\max\{\underset{\sim}{a}_2^{1ave}, \underset{\sim}{r}_1^{ave}\} + s_{31}^1 + \frac{\underset{\sim}{ps}_1^{1\,ave}}{v_{21}^1}$ = 262.830;

Again, job 1 is assigned to machine 1, since the fuzzy completion time is shorter in this case. All results are shown in Table 7.

Table 7 Fuzzy completion times at stage 1.

Job #	Machine #	Fuzzy completion times	Average fuzzy completion times
3	2	(134.000, 141.000, 145.000)	140.000
1	1	(126.138, 134.088, 142.922)	134.383
5	2	(217.261, 224.261, 233.706)	225.076
4	1	(231.770, 247.630, 261.736)	247.045
2	1	(289.770, 312.409, 329.905)	310.695

Part 4: Assigning the jobs to the machines at the other stages
Step 8: Find the job sequence of the next stage.

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

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

Step 9: Again as in Step 5, assign the first job $j_{[1]}$ (job 3 in case 1 and job 1 in case 2) to the selected 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 job 3 is assigned to the machine first, whereas job 1 is assigned to the machine first otherwise.
Step 10: Update the availability of the selected machine by using the value of the average fuzzy completion time of the job assigned to the machine.
Step 11: Remove job 3 in case 1 (or job 1 in case 2) 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 FIFO rule generates a solution which is better than that generated by the permutation rule, so we select the solution generated by the FIFO rule and the fuzzy completion time is (591.831, 621.073, 651.315).

Part 5: Finding the best solution
Step 14: Repeat Steps 2 to Step 14 again for the other representatives, and return the best fuzzy solution. The results for this example are shown in Table 9.

Table 8 Fuzzy completion times at stage 2.

Job #	Machine #	Fuzzy completion times	Average fuzzy completion times
Case 1: Permutation rule			
3	1	(209.770, 218.884, 231.340)	219.998
1	1	(314.947, 330.212, 347.941)	331.033
5	1	(401.228, 417.551, 435.280)	418.020
4	1	(512.741, 532.471, 555.310)	533.507
2	1	(604.693, 632.984, 658.393)	632.023
Case 2: FIFO rule			
1	1	(220.315, 234.417, 248.523)	234.418
3	1	(289.085, 305.301, 327.864)	307.417
5	1	(388.366, 405.639, 428.202)	407.402
4	1	(499.879, 520.559, 548.233)	522.890
2	1	(591.831, 621.073, 651.315)	621.406

Table 9 The best fuzzy solution.

Job #	Machine #	Fuzzy completion times	Average fuzzy completion times
Stage1:			
4	2	(103.571, 110.559, 115.217)	109.782
3	1	(178.646, 188.561, 194.227)	187.145
1	2	(213.264, 230.991, 247.583)	230.613
5	2	(276.525, 294.252, 316.289)	295.689
2	1	(264.646, 281.341, 290.396)	278.794
Stage2:			
4	1	(196.513, 206.479, 216.248)	206.413
3	1	(261.282, 273.363, 291.588)	275.411
1	1	(366.460, 384.691, 408.189)	386.447
5	1	(452.741, 472.030, 495.528)	473.433
2	1	(505.693, 533.543, 559.610)	532.949

Table 10 An optimal solution.

Case of $\underset{\sim}{ps}_j^t$	Completion times
$a_{\underset{\sim}{ps}_j^t}$	474.693
$b_{\underset{\sim}{ps}_j^t}$	512.543
$c_{\underset{\sim}{ps}_j^t}$	528.609
$\underset{\sim}{ps}_j^{t\ ave}$	501.807

For an optimal solution, we have used the standard processing times by using the values $a_{\underset{\sim}{ps}_j^t}$, $b_{\underset{\sim}{ps}_j^t}$, $c_{\underset{\sim}{ps}_j^t}$, and $\underset{\sim}{ps}_j^{t\ ave}$ 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.

5. Computational Results

In our tests, we used problems with 5, 10, 20, and 100 jobs, 2 and 5 machines per stage, and 2 and10 stages. For each problem size, ten different instances have been run. The standard processing times are fuzzy such that the values of $b_{\underset{\sim}{ps}_j^t}$ are generated uniformly from the interval [10,100], $a_{\underset{\sim}{ps}_j^t} = b_{\underset{\sim}{ps}_j^t}$ - 10×U[0,1] and $c_{\underset{\sim}{ps}_j^t} = b_{\underset{\sim}{ps}_j^t}$ + 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 zero and half of their total standard processing time mean.

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

Algorithms		$\underset{\sim}{C}_{max}$	%deviation
FLPT^T	*a*	306.480	14.108
	b	327.611	15.467
	c	346.916	14.954
	ave	327.002	15.143
FLPT^1	*a*	328.988	22.488
	b	348.015	22.659
	c	367.626	21.816
	ave	348.210	22.610
FLPT^2	*a*	282.449	5.160
	b	297.872	4.986
	c	315.784	4.638
	ave	298.701	5.178

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

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}	Average CPU Times (ms)
(5/2/2)	327.002	348.210	**298.701**									0.667
(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	3.818
(10/2/2)	491.604	520.869	**478.012**									2.667
(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**	6.182
(10/5/2)	245.055	269.105	**240.404**									2.333
(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**	11.819
(20/2/2)	933.492	1022.701	**892.706**									2.333
(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**	11.182
(20/5/2)	403.685	454.101	**387.865**									7.333
(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**	22.636
(100/2/2)	4042.877	4488.094	**3951.540**									15.000
(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**	49.545
(100/5/2)	1555.362	1743.006	**1511.234**									24.333
(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**	104.273

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

First, we present the results of the fuzzy LPT 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 in an average CPU time of 5.993 seconds and the average makespan values of the fuzzy LPT algorithms as shown in Table 11. The average optimal makespan values using the fuzzy standard processing time values $a_{\underset{\sim}{ps}_j^t}$, $b_{\underset{\sim}{ps}_j^t}$, $c_{\underset{\sim}{ps}_j^t}$, and $\underset{\sim}{ps}_j^{t\ ave}$ are 268.589, 283.727, 301.787, and 283.997, respectively. It can be observed that 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.

Next, for the large-size problems, we cannot find an optimal solution within a reasonable time since these problems are NP hard [10]. Thus, we used fuzzy LPT 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 and the average CPU times (in milliseconds). The results show that a fuzzy LPT algorithm by sorting the jobs in descending order of the average values of the total operating times $\underset{\sim}{t}_j^{/k}$ (i.e. an FLPT^k algorithm) gives good solutions. Although an FLPT^5 algorithm outperforms the others for the test problems with 5 jobs, 2 machines per stage, and 10 stages, its value is only slightly better than the value of an FLPT^{10} algorithm. In general, the quality of an FLPT^t algorithm improves with an increasing value of *t*.

6. Conclusions

In this paper, fuzzy LPT algorithms 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 LPT algorithms are proposed to manage jobs with uncertain standard processing times. This approach generates a scheduling result with a membership function completion time. 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. In particular, an FLPT^k algorithm that uses the average values of the total operating times gives good solutions for both small- and large-size test problems.

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.

Acknowledgments

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

[5] 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.

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

[7] Ross, T.J. 1995. Fuzzy logic with engineering applications, McGraw-Hill, New York, U.S.

[8] 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.

[9] 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, 1998: 97-101.

[10] Gupta, J.N.D. 1988. Two-stage, hybrid flowshop scheduling problem. Journal of the Operational Research Society, 39(4): 359-364.

IE Network Conference 2007

24-26 October 2007
Royal Phuket City Hotel, Phuket