

## Short Paper

---

# Fuzzy Palmer Scheduling for Flow Shops with More Than Two Machines

TZUNG-PEI HONG\* AND TZUNG-NAN CHUANG

*Department of Information Management*

*I-Shou University*

*Kaohsiung, Taiwan 840, R.O.C.*

*\*E-mail: tphong@csa500.isu.edu.tw*

*E-mail: tnchuang@mail.ntu.edu.tw*

In the past, we have demonstrated how fuzzy concepts can easily be used in the Johnson algorithm to manage uncertain scheduling on two-machine flow shops. This paper extends application to fuzzy flow shops with more than two machines. A new fuzzy heuristic flow-shop scheduling algorithm (the fuzzy Palmer algorithm) is then designed since optimal solutions seem unnecessary for uncertain environments. Also, the conventional Palmer algorithm is presented as a special case of the fuzzy Palmer algorithm with special assigned membership functions.

**Keywords:** Palmer algorithm, completion time, flow shop, fuzzy task, scheduling

## 1. INTRODUCTION

In a flow shop [14], jobs are processed by a series of machines in a predefined order. For jobs scheduled in a two-machine flow shop, Johnson proposed an algorithm to achieve a minimum makespan [8]. For jobs scheduled in a flow shop of more than two machines, no optimal solution except exhaustive search has ever been proposed. Palmer proposed a heuristic method to achieve a nearly minimum makespan [13].

In the past, the processing time for each job has usually been assigned or estimated as a fixed value. In many real-world applications, however, the processing time for each job may vary dynamically with the situation. Several theories, such as *fuzzy set theory*, *probability theory*, *D-S theory*, and approaches based on *certainty factors*, have been developed to manage uncertainty. Among them, fuzzy set theory is more and more frequently used in intelligent control because of its simplicity and similarity to human reasoning. Although fuzzy set concepts are mainly used in linguistic domains, they are also used in numerical domains, where each number is assigned a membership value. Examples are Chanas and Kolodziejczyk's fuzzy network flow capacity [1, 2], Gazdik's fuzzy network planning [4], Han *et al.*'s fuzzy due-date scheduling [5], Hong *et al.*'s Fuzzy LPT scheduling [6], Klein's

---

Received November 4, 1997; revised May 14, 1998 & July 29, 1998; accepted August 3, 1998.  
Communicated by Shing-Tsaan Huang.

fuzzy shortest path [9], Nasution's fuzzy critical path [12], McCahon and Lee's fuzzy project network analysis [10] and fuzzy job sequencing [11], Chang *et al.*'s fuzzy project planning [3], and so on.

In [7], we used fuzzy concepts in Johnson's algorithm for two-machine fuzzy scheduling. In this paper, we attempt to generalize this idea to the Palmer algorithm for flow shops with more than two machines. Since the processing time for each task may vary dynamically and may involve uncertainty in some real-world applications, optimal solutions seem unnecessary, and the Palmer algorithm is thus sufficient for uncertain environments. Also, the conventional Palmer algorithm will be presented as a special case of the fuzzy Palmer algorithm with special assigned membership functions. The fuzzy Palmer algorithm is then a feasible solution for both deterministic and uncertain scheduling in flow shops with more than two machines.

## 2. REVIEW OF THE PALMER HEURISTIC ALGORITHM

We will now state an  $m$ -machine ( $m > 2$ ) flow-shop problem with a makespan criterion. Given a set of  $n$  independent jobs, each having  $m$  tasks ( $T_{11}, T_{21}, \dots, T_{m1}, T_{12}, T_{22}, \dots, T_{(m-1)n}, T_{mn}$ ) that must be executed in the same sequence on  $m$  machines ( $P_1, P_2, \dots, P_m$ ), scheduling seeks the minimum completion time for the last job. Since this problem is NP-hard, Palmer proposed the following heuristic algorithm to solve it in polynomial time [13]:

### The Palmer heuristic algorithm.

**Input:** A set of  $n$  jobs, each having  $m$  ( $m > 2$ ) tasks executed, respectively, on each of  $m$  machines.

**Output:** A schedule with a nearly minimum completion time of the last job.

**Step 1:** For each job  $J_j$ , find the value  $\pi_j$  as follows:

$$\pi_j = \sum_{i=1}^{\lceil m/2 \rceil} -(m-2i+1)t_{ij} + (m-2i+1)t_{(m+1-i)j}.$$

**Step 2:** Sort the jobs in descending order of  $\pi_j$ 's; if two or more jobs have the same value of  $\pi_j$ , then sort them in an arbitrary order.

**Step 3:** Schedule the jobs on the machines in the sorted order.

## 3. THE FUZZY PALMER SCHEDULING ALGORITHM

In the fuzzy Palmer scheduling algorithm, each processed task is a fuzzy set. Fuzzy operations are then used to schedule uncertain jobs and to find the fuzzy completion time. The fuzzy Palmer scheduling algorithm using the fuzzy averaging ranking method is stated below. Note that other ranking methods can also be used in our fuzzy scheduling algorithm.

### 3.1 Notation

Notations used in this paper are listed below.

- $n$ : the number of jobs;
- $m$ : the number of machines;
- $T_{ij}$ : the  $i$ -th task for the  $j$ -th job,  $i = 1, 2, \dots, m$  and  $j = 1, 2, \dots, n$ ;
- $t_{ij}$ : the fuzzy execution time (a fuzzy set) of  $T_{ij}$ ;
- $t_{ijk}$ : the  $k$ -th possible execution time of  $t_{ij}$ ,  $1 \leq k \leq |\text{supp}(t_{ij})|$ , where  $\text{supp}(t_{ij})$  is the support of the fuzzy set  $t_{ij}$ ;
- $\mu(t_{ijk})$ : the membership value of  $t_{ijk}$ ;
- $P_k$ : the  $k$ -th machine,  $k = 1, 2, \dots, m$ ;
- $p_k$ : the fuzzy execution time (a fuzzy set) of  $P_k$ ;
- $p_{kj}$ : the  $j$ -th possible execution time of  $p_k$ ,  $1 \leq j \leq |\text{supp}(p_k)|$ ;
- $\mu(p_{kj})$ : the membership value of  $p_{kj}$ ;
- $f$ : the final completion time (a fuzzy set) of the whole schedule.

### 3.2 The Fuzzy Palmer Scheduling Algorithm

The proposed fuzzy Palmer scheduling algorithm is stated as follows.

#### The fuzzy Palmer scheduling algorithm.

**Input:** A set of  $n$  jobs, each with  $m$  tasks to be executed, respectively, on each of  $m$  machines; each task has a processing time membership function.

**Output:** A fuzzy schedule with a completion time membership function  $f$ .

**Step 1:** For each job  $J_j$ , find the value  $\pi_j$  using the fuzzy addition operation as follows:

$$\pi_j = \sum_{i=1}^{\lceil m/2 \rceil} -(m-2i+1)t_{ij} + (m-2i+1)t_{(m+1-i)j}.$$

**Step 2:** For each  $\pi_j$ , find its average value  $\pi_j^{ave}$  using the following formula:

$$\pi_j^{ave} = \frac{\sum_{k=1}^{|\text{supp}(\pi_j)|} (\mu(\pi_{jk}) \times \pi_{jk})}{\sum_{k=1}^{|\text{supp}(\pi_j)|} \mu(\pi_{jk})}$$

where  $\pi_{jk}$  is the  $k$ -th possible value of  $\pi_j$ .

**Step 3:** Sort the jobs in descending order of  $\pi_j^{ave}$ 's if two or more jobs have the same value of  $\pi_j^{ave}$ , then sort them in an arbitrary order.

**Step 4:** Set the initial completion time  $p_1, p_2, \dots, p_m$  for machines  $P_1, P_2, \dots, P_m$  to zero with a fuzzy value of 1.

**Step 5:** Schedule the first job  $J_j$  in the sorted list to the machines such that  $T_{1j}$  is assigned to  $P_1$ ,  $T_{2j}$  is assigned to  $P_2$ , ..., and  $T_{mj}$  is assigned to  $P_m$ .

**Step 6:** Set  $p_1 = p_1 + t_{1j}$  using the fuzzy addition operation.

**Step 7:** Set  $p_{(i+1)} = \text{find-longer-time}(p_i, p_{(i+1)}) + t_{(i+1)j}$ ,  $i = 1, 2, \dots, (m-1)$ , where the find-longer-time procedure is used to find the fuzzy start time for machine  $P_{(i+1)}$ .

**Step 8:** Remove task  $J_j$  from the sorted list.

**Step 9:** Repeat Steps 5 to 8 until the sorted list is empty.

**Step 10:** Set the final completion time  $f = p_m$ .

After Step 10, scheduling is finished, and a fuzzy completion time with a membership function  $f$  has been found. The *find-longer-time* Procedure, which is similar to that in [7], is stated as follows.

### The Find-longer-time Procedure.

**Input:** Two fuzzy sets with completion times,  $p_r$  and  $p_{r+1}$ , for each machine.

**Output:** The start time  $S$  (a fuzzy set) for the next job to be executed on machine  $r+1$ .

**Procedure:**

- (a1) FOR each fuzzy set  $p_k$ ,  $k = r$  and  $r+1$ , DO the following:
  - (b1) Sort  $p_k$  in descending order of  $p_{kj}$ 's, where  $1 \leq j \leq |\text{supp}(p_k)|$ .
  - (b2) Build a new fuzzy set  $l_k$  with a term  $\mu(l_{k_0})/l_{k_0}$  added at the front of it, and set  $l_{k_0} = \infty$  and  $\mu(l_{k_0}) = 0$  as a starting point.
  - (b3) Set each other element  $l_{kj} = p_{kj}$  and  $\mu(l_{kj}) = \max[\mu(p_{kj}), \mu(l_{k(j-1)})]$ .
 END FOR
- (a2) Merge the  $p_k$ 's ( $k = r$  and  $r+1$ ) into a fuzzy list  $S$  in a descending order of  $p_{kj}$ 's; the label  $P_k$  is also attached to each element to represent its source set, that is,
 
$$S = \{(\mu(s_1)/s_1, P_{h(s_1)}), (\mu(s_2)/s_2, P_{h(s_2)}), \dots, (\mu(s_{|\text{supp}(p_r)|+|\text{supp}(p_{r+1})|})/s_{|\text{supp}(p_r)|+|\text{supp}(p_{r+1})|}, P_{h(s_{|\text{supp}(p_r)|+|\text{supp}(p_{r+1})|)}})\},$$
 where  $P_{h(s_j)}$  denotes the source set of  $s_j$ .
- (a3) Attach a pointer to the first element  $\mu(l_{k_0})/l_{k_0}$  for each  $l_k$ ,  $k = r$  and  $r+1$ .
- (a4) FOR each element  $(\mu(s_i)/s_i, P_{h(s_i)})$  in  $S$ ,  $i = 1$  to  $|\text{supp}(p_r)| + |\text{supp}(p_{r+1})|$ , DO the following:
  - (c1) Process  $l_k$  such that  $k \neq h(s_i)$  is satisfied as follows:
    - (d1) get the element which the current pointer points to;
    - (d2) move the pointer to the element  $\mu(l_{k_j})/l_{k_j}$ , where  $l_{k_j} > s_i$  and  $l_{k_{(j+1)}} \leq s_i$ ;
  - (c2) Set  $\mu(s_i) = \text{Min}[\mu(s_i), 1 - \mu(l_{k_j})]$ ;
  - (c3) IF  $s_i = s_{i-1}$ ,  
 THEN set  $\mu(s_i) = \text{Max}[\mu(s_i), \mu(s_{i-1})]$  and remove the term  $(\mu(s_{i-1})/s_{i-1}, P_{h(s_{i-1})})$  from  $S$ .
 END FOR
- (a5) Normalize and output the final  $S$  in ascending order of  $s_i$  (without the  $P_{h(s_j)}$  subterm).

After Step (a5),  $S$ , the start time for the next job to be executed on machine  $r+1$ , has been found. Details of this procedure can be found in [7]. The following example shows how the fuzzy Palmer algorithm works.

### 3.3 An Example

Assume a group of five jobs is to be processed in a three-step operation. The first step is degreasing, the second is painting, and the third is drying. Assume also the fuzzy execution times of these jobs are as listed in Table 1.

**Table 1. The processing time of the five jobs in the example.**

| Job   | $t_{1j}$       | $t_{2j}$       | $t_{3j}$       |
|-------|----------------|----------------|----------------|
| $J_1$ | {1.0/4}        | {1.0/7}        | {1.0/3, 0.9/4} |
| $J_2$ | {0.5/4, 1.0/5} | {1.0/5}        | {1.0/6}        |
| $J_3$ | {1.0/5, 0.9/6} | {1.0/2, 0.8/3} | {1.0/4}        |
| $J_4$ | {1.0/1}        | {0.9/4, 1.0/5} | {1.0/2, 0.9/3} |
| $J_5$ | {1.0/2, 0.2/4} | {1.0/5}        | {0.7/2, 1.0/3} |

Using the fuzzy Palmer scheduling algorithm, the execution process is as follows.

$$\begin{aligned} \text{Step 1: Set } \pi_j &= \sum_{i=1}^{\lceil 3/2 \rceil} -(3-2i+1)t_{ij} + (3-2i+1)t_{(3+1-i)j} \\ &= -2t_{1j} + 2t_{3j}. \end{aligned}$$

**Step 2:** For each  $\pi_j$ , find the average value  $\pi_j^{ave}$ . The results are shown in Table 2.

**Table 2. The results of  $\pi_j^{ave}$  for this example.**

| Jobj          | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ |
|---------------|-------|-------|-------|-------|-------|
| $\pi_j^{ave}$ | -1.1  | 2.7   | -2.9  | 2.9   | 0.4   |

**Step 3:** The complete sequence is  $\{J_4, J_2, J_5, J_1, J_3\}$ .

**Step 4:** Set  $p_1 = p_2 = p_3 = \{1.0/0\}$ .

**Step 5:** Assign  $J_4$  (the first job in the sorted order) to machines for execution.

**Step 6:** Set  $p_1 = p_1 + t_{14} = \{1.0/0\} + \{1.0/1\} = \{1.0/1\}$ .

**Step 7:** Set  $p_2 = \text{find-longer-time}(p_1, p_2) + t_{24}$   
 $= \{1.0/1\} + \{0.9/4, 1.0/5\}$   
 $= \{0.9/5, 1.0/6\};$

Set  $p_3 = \text{find-longer-time}(p_2, p_3) + t_{34}$   
 $= \{0.9/5, 1.0/6\} + \{1.0/2, 0.9/3\}$   
 $= \{0.9/7, 1.0/8, 0.9/9\}.$

**Step 8:** Remove  $J_4$ .

**Step 9:** Repeat Steps 5 to 8 for the other jobs. The final results are shown in Table 3 and Fig. 1.

**Step 10:** The final completion time  $f = p_3 = \{0.2/29, 1.0/30, 0.9/31\}$ .

**Table 3. The scheduling results for this example**

| Job $j$ | $p_1$ (after $t_{1j}$ is added)      | $p_2$ (after $t_{2j}$ is added) | $p_3$ (after $t_{3j}$ is added) |
|---------|--------------------------------------|---------------------------------|---------------------------------|
| $J_4$   | {1.0/1}                              | {0.9/5,1.0/6}                   | {0.9/7,1.0/8,0.9/9}             |
| $J_2$   | {0.5/5,1.0/6}                        | {0.9/10,1.0/11}                 | {0.9/16,1.0/17}                 |
| $J_5$   | {0.5/7,1.0/8,0.2/9,0.2/10}           | {0.9/15,1.0/16}                 | {0.7/18,0.9/19,1.0/20}          |
| $J_1$   | {0.5/11,1.0/12,0.2/13,0.2/14}        | {0.9/22,1.0/23}                 | {0.9/25,1.0/26,0.9/27}          |
| $J_3$   | {0.5/16,1.0/17,0.9/18,0.2/19,0.2/20} | {0.9/24,1.0/25,0.8/26}          | {0.2/29,1.0/30,0.9/31}          |

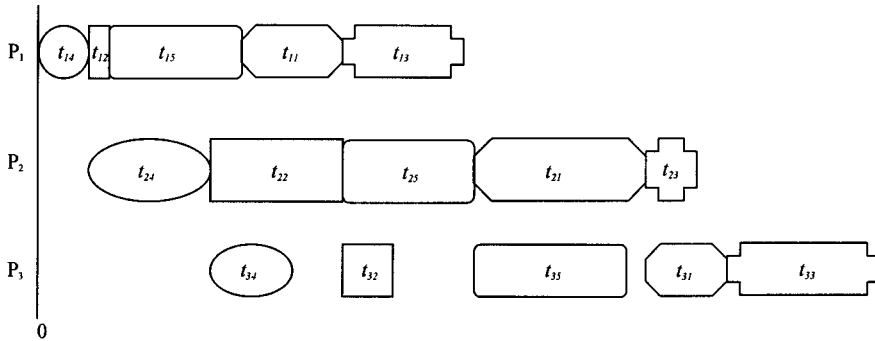


Fig. 1. The final schedule results for this example.

#### 4. REDUCING TO THE ORIGINAL PALMER SCHEDULING ALGORITHM

In this section, the fuzzy Palmer scheduling algorithm is shown to be equivalent to the original Palmer scheduling algorithm in environments where the latter is applied. For the original Palmer scheduling algorithm to work, the task time must be known and definite. That is, each task must have a time with a single membership value = 1, and all other time membership values = 0. In this environment, the two algorithms can be proven to be equivalent as follows.

**Theorem 1:** If Fuzzy set  $A = \{1/a\}$  and Fuzzy set  $B = \{1/b\}$ , then  $\text{find-longer-time}(A, B) = \{1/\max(a, b)\}$ .

**Proof:** This has been proven in [7].

**Theorem 2:** The execution sequences using the fuzzy Palmer algorithm and using the original Palmer algorithm are the same for definite tasks if, when there is a tie, the rules that allocate jobs to the processors are the same.

**Proof:** Let  $ot_{kj}$  and  $ft_{kj}$  denote, respectively, the definite execution time in the original Palmer algorithm and the fuzzy execution time in the fuzzy Palmer algorithm for task  $T_{kj}$ . Since the time for each task  $T_{kj}$  is definite, the fuzzy execution time  $ft_{kj} = \{1/ot_{kj}\}$ . Then,

$$\begin{aligned}
f\pi_j &= \sum_{i=1}^{\lceil m/2 \rceil} -(m-2i+1)ft_{ij} + (m-2i+1)ft_{(m+1-i)j} \\
&= \sum_{i=1}^{\lceil m/2 \rceil} -(m-2i+1)\{1/ot_{ij}\} + (m-2i+1)\{1/ot_{(m+1-i)j}\} \\
&= \{1/ \sum_{i=1}^{\lceil m/2 \rceil} (-(m-2i+1) \times ot_{ij} + (m-2i+1) \times ot_{(m+1-i)j})\} \\
&= \{1/ o\pi_j\}.
\end{aligned}$$

This implies that

$$f\pi_j^{ave} = o\pi_j.$$

The sorted job list resulting from Steps 1 to 3 of the fuzzy Palmer algorithm is then the same as that yielded by the original Palmer scheduling algorithm. Since the tasks are scheduled according to the sorted list in both algorithms, the proof is complete.  $\square$

**Theorem 3:** If the schedule time from the original Palmer scheduling algorithm is  $df$ , then the fuzzy schedule time  $ff$  from the fuzzy Palmer scheduling algorithm is  $\{1/df\}$ .

**Proof:** Without loss of generality, we may assume that the final scheduling order is  $J_1$  to  $J_m$  using both algorithms (from Theorem 2). We will prove by induction that the intermediate scheduling results yielded by the two scheduling algorithms are the same. First, the correctness of the theorem for  $J_1$  is proven. In this case, the original Palmer scheduling algorithm initially sets the execution time ( $op_1$  to  $op_m$ ) for  $P_1$  to  $P_m$  to 0, and the fuzzy Palmer scheduling algorithm initially sets the fuzzy execution time  $fp_k = \{1/0\} = \{1/op_k\}$ , for  $k=1$  to  $m$ . The first job  $J_1$  is then put into the processors by the two algorithms. The execution time  $op_1$  yielded by the original Palmer scheduling algorithm becomes

$$\begin{aligned}
new\ op_1 &= old\ op_1 + ot_{11} \\
&= 0 + ot_{11} \\
&= ot_{11}.
\end{aligned}$$

The execution time  $op_2$  yielded by the original Palmer scheduling algorithm becomes

$$\begin{aligned}
new\ op_2 &= Max(new\ op_1, old\ op_2) + ot_{21} \\
&= Max(ot_{11}, 0) + ot_{21} \\
&= ot_{11} + ot_{21}.
\end{aligned}$$

In general, we can get

$$\begin{aligned}
new\ op_k &= Max(new\ op_{k-1}, old\ op_k) + ot_{k1} \\
&= \sum_{l=1}^k ot_{l1}.
\end{aligned}$$

The execution time,  $fp_1$ , yielded by the fuzzy Palmer scheduling algorithm becomes

$$\begin{aligned}
new\ fp_1 &= old\ fp_1 + ft_{11} \\
&= \{1/0\} + \{1/ot_{11}\} \\
&= \{1/ot_{11}\} \\
&= \{1/new\ op_1\}.
\end{aligned}$$

The execution time,  $fp_2$ , yielded by the fuzzy Palmer scheduling algorithm becomes

$$\begin{aligned}
new\ fp_2 &= find-longer-time(new\ fp_1, old\ fp_2) + ft_{21} \\
&= find-longer-time(1/new\ op_1, 1/0) + ft_{21} \\
&= \{1/Max(new\ op_1, 0)\} + \{1/ot_{21}\} \quad (\text{from Theorem 1}) \\
&= \{1/(new\ op_1 + ot_{21})\} \\
&= \{1/(ot_{11} + ot_{21})\} \\
&= \{1/new\ op_2\}.
\end{aligned}$$

In general, we can get

$$\begin{aligned}
new\ fp_k &= find-longer-time(new\ fp_{k-1}, old\ fp_k) + ft_{k1} \\
&= \{1 / \sum_{l=1}^k ot_{l1}\} \\
&= \{1/new\ op_k\}.
\end{aligned}$$

Next, by induction, we assume that  $fp_k$  is  $\{1/op_k\}$  for  $k = 1$  to  $m$  after  $J_s$  is scheduled and prove that the results are also valid for  $J_{s+1}$ . The execution time  $op_1$  yielded by the original Palmer scheduling algorithm becomes

$$new\ op_1 = old\ op_1 + ot_{1(s+1)}.$$

The execution time  $op_2$  yielded by the original Palmer scheduling algorithm becomes

$$new\ op_2 = Max(new\ op_1, old\ op_2) + ot_{2(s+1)}.$$

In general, we can get

$$new\ op_k = Max(new\ op_{k-1}, old\ op_k) + ot_{k(s+1)}.$$

The execution time,  $fp_1$ , yielded by the fuzzy Palmer scheduling algorithm becomes

$$\begin{aligned}
new\ fp_1 &= old\ fp_1 + ft_{1(s+1)} \\
&= \{1/old\ op_1\} + \{1/ot_{1(s+1)}\} \\
&= \{1/(old\ op_1 + ot_{1(s+1)})\} \\
&= \{1/new\ op_1\}.
\end{aligned}$$

The execution time,  $fp_2$ , yielded by the fuzzy Palmer scheduling algorithm becomes

$$\begin{aligned}
new\ fp_2 &= find-longer-time(new\ fp_1, old\ fp_2) + ft_{2(s+1)} \\
&= find-longer-time(\{1/new\ op_1\}, \{1/old\ op_2\}) + ft_{2(s+1)} \\
&= \{1/Max(new\ op_1, old\ op_2)\} + \{1/ot_{2(s+1)}\} \quad (\text{from Theorem 1}) \\
&= \{1/(Max(new\ op_1, old\ op_2) + ot_{2(s+1)})\} \\
&= \{1/new\ op_2\}.
\end{aligned}$$

In general, we can get

$$\begin{aligned}
 new\ fp_k &= find-longer-time(new\ fp_{k-1}, old\ fp_k) + ft_{k(s+1)} \\
 &= find-longer-time(\{1/new\ op_{k-1}\}, \{1/old\ op_k\}) + ft_{k(s+1)} \\
 &= \{1/Max(new\ op_{k-1}, old\ op_k)\} + \{1/ot_{k(s+1)}\} \\
 &= \{1/Max(new\ op_{k-1}, old\ op_k) + ot_{k(s+1)}\} \\
 &= \{1/new\ op_k\}.
 \end{aligned}$$

Therefore, after all jobs are scheduled,  $fp_m = \{1/op_m\}$ . According to Step 10 of the fuzzy Palmer scheduling algorithm, the completion time  $ff$  is  $fp_m$ . According to the original Palmer scheduling algorithm, the completion time  $df$  is  $op_m$ . So,  $ff = fp_m = \{1/op_m\} = \{1/df\}$ . This completes the proof.  $\square$

## 5. CONCLUSIONS

In this paper, we have proposed the fuzzy Palmer algorithm for scheduling uncertain jobs in a flow shop with more than two machines. The fuzzy Palmer algorithm can yield a scheduling result with a membership function for the final completion time. The results can then help managers gain a broader overall view of scheduling. Also, the conventional Palmer algorithm has been shown to be a special case of the fuzzy Palmer algorithm with special assigned membership functions. The fuzzy Palmer algorithm is then a feasible solution for both deterministic and uncertain flow shops with more than two machines. In the future, we will try to apply other characteristics of fuzzy sets to the scheduling field.

## ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their very constructive comments.

## REFERENCES

1. S. Chanas and W. Kolodziejczyk, "Real-value flows in a network with fuzzy arc capacities," *Fuzzy Sets and Systems*, Vol. 13, No. 2, 1984, pp. 139-151.
2. S. Chanas and W. Kolodziejczyk, "Integer flows in network for fuzzy capacity constraints," *Networks*, Vol. 16, No.1, 1986, pp. 17-32.
3. I. S. Chang, Y. Tsujimura, M. Gen and T. Tozawa, "An efficient approach for large scale project planning based on fuzzy Delphi method," *Fuzzy Sets and Systems*, Vol. 76, No. 3, 1995, pp. 277-288.
4. I. Gazdik, "Fuzzy-network planning," *IEEE Transactions on Reliability*, Vol. 32, No. 3, 1983, pp. 304-313.
5. S. Han, H. Ishii and S. Fujii, "One machine scheduling problem with fuzzy duedates," *European Journal of Operational Research*, Vol. 79, No. 1, 1994, pp. 1-12.
6. T. P. Hong, C. M. Huang and K. M. Yu, "LPT scheduling for fuzzy tasks," *Fuzzy Sets and Systems*, Vol. 97, No. 3, 1998, pp. 277-286.

7. T. P. Hong and T. N. Chuang, "Fuzzy scheduling on two-machine flow shop," *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, Vol. 6, No. 4, 1998, pp. 471-481.
8. S. M. Johnson, "Optimal two- and three-stage production scheduling with setup times included," *Naval Research Logistics Quarterly*, Vol. 1, No. 1, 1954, pp. 61-68.
9. C. M. Klein, "Fuzzy shortest paths," *Fuzzy Sets and Systems*, Vol. 39, No. 1, 1991, pp. 27-41.
10. C. S. McCahon and E. S. Lee, "Project network analysis with fuzzy activity time," *Computer Mathematic Application*, Vol. 15, No. 10, 1988, pp. 829-838.
11. C. S. McCahon and E. S. Lee, "Job sequencing with fuzzy processing times," *Computer Mathematic Application*, Vol. 19, No. 7, 1990, pp. 31-41.
12. S. H. Nasution, "Fuzzy critical path method," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 1, 1994, pp. 48-57.
13. D. S. Palmer, "Sequencing jobs through a multi-stage process in the minimum total time- a quick method of obtaining a near optimum," *Operational Research Quarterly*, Vol. 16, No. 3, 1965, pp. 101-107.
14. M. Pinedo, *Scheduling Theory, Algorithms, and Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

**Tzung-Pei Hong (洪宗貝)** received his B. S. degree in chemical engineering from National Taiwan University in 1985, and his Ph.D. degree in computer science and information engineering from National Chiao Tung University in 1992.

From 1987 to 1994, he was with the Laboratory of Knowledge Engineering, National Chiao Tung University, where he was involved in applying techniques of parallel processing to artificial intelligence. From 1992 to 1994, he was an associate professor in the Department of Computer Science at Chung-Hua Polytechnic Institute. He is currently an associate professor in the Department of Information Management at I-Shou University and an associate researcher at the National University of Kaohsiung in Preparation. His current research interests include parallel processing, machine learning, neural networks, fuzzy sets, expert systems, campus computerization, and management information systems.

Dr. Hong is a member of the Association for Computing Machinery, the IEEE Computer Society, the Chinese Fuzzy Systems Association and the Institute of Information and Computing Machinery. He is also included in the Marquis 16th Edition of Who's Who in the World.

**Tzung-Nan Chuang (莊宗南)** received his B. S. degree in industrial engineering from Tung-Hai University in 1994, and his MBA degree in management science from I-Shou University in 1996.

From 1994 to 1998, he was with the Laboratory of Management Science at I-Shou University, where he was involved in applying fuzzy set theories to scheduling. He is currently a Ph. D. student in the Department of Business Administration at National Cheng-Kung University. His current research interests include fuzzy sets, scheduling, logistics, and management information systems.