ISSN 1 746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 2 (2006) No. 3, pp. 160-166

A class of fuzzy random programming and its application to parallel machine scheduling*

Yang Wang , Jiuping Xu†

Uncertainty Decision-Making Laboratory
School of Business and Administration, Sichuan University, Chengdu 610064, P. R. China

(*Received January 8 2006, Accepted June 1 2006*)

Abstract. In many basic scheduling models, jobs' processing times and due-dates are deterministic. Practically, they are as uncertain variables in some more models. Furthermore, in a real situation of decision making, there exists uncertainty that can not be described only by fuzziness but also randomness. The purpose of this paper is to develop a methodology for parallel machines scheduling problem with fuzzy random due-dates. One type of fuzzy random scheduling models is present. Finally, algorithm and application example are given.

Keywords: fuzzy random programming, parallel machine scheduling model, fuzzy random due-dates

1 Introduction

Recently, scheduling theory has drawn much attention. In decision-making area, scheduling is very important in manufacturing and in service industries. A scheduling problem deals with an optimal allocation of a limited number of given resources (such as machines) to given tasks (such as jobs) over time. Modern scheduling theory contains two main parts, referring to deterministic and uncertain (stochastic, fuzzy) models. Despite that a huge number of papers and books published about scheduling, the utilization of numerous theoretical results of classical scheduling theory in most production environments is far away from the desired plan.

In fact, various factors involved in the scheduling problems are often imprecise or uncertain more or less in nature when we formulate scheduling problems in the real-world. This is especially obvious when human-made factors are added to the problems. In these cases, it seems more appropriate to consider fuzzy processing times, fuzzy due-dates, and so on. So far, much of research work has been performed on fuzzy scheduling problems. The earliest paper in fuzzy scheduling appeared in 1979[19]. Ishii et al.[5] studied scheduling problems with fuzzy due-dates. Han et al.[3] considered single-machine scheduling problem with fuzzy due-dates. And then, Ishibuchi et al.[4] studied flow shop scheduling with fuzzy processing times. The fuzzy job shop scheduling problem was investigated by Kuroda and Wang[9]. Konno and Ishii[8] discussed an open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint. Dubois et al.[2] formulated a simple mathematical model of job-shop scheduling under preference and uncertainty and outlined a combinatorial search method to solve the model. McCahon and Lee[18] modified Johnsons algorithm and Ignall and Schrages branch and bound algorithm to accept triangular and trapezoidal fuzzy processing times. Hong et al.[20] utilized fuzzy set concept in the largest processing time (LPT) algorithm to schedule fuzzy tasks. Recently, Itoh and Ishii[6] proposed a single machine scheduling model dealing with fuzzy processing times

* This research was supported by the National Science Foundation for Distinguished Young Scholars, P. R. China (Grant No. 70425005) and the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE of P. R. China (Grant No. 20023834-3).

† Corresponding author. Tel: +86-28-85418522; fax: +86-28-85400222.
E-mail address: xujiuping@openmba.com, leyunyang@126.com.

and due-dates by the possibility measure. Litoiu and Tadei[10] presented some new models for real-time task scheduling with fuzzy deadlines and processing times. There are three main approaches reported in the literature for the fuzzy scheduling problems: using fuzzy ranking and fuzzy dominance relation methods, and solving mathematical programming models to determine the optimal schedules by heuristic approximation methods including genetic algorithm (GA), simulated annealing, tabu search[1], etc.

A limited amount of literature has been devoted to fuzzy random parallel machine scheduling problems. In this paper, as a practical application, we focus on the model with fuzzy random due-dates. We design a genetic algorithm to solve the formulated scheduling models. Effectiveness of the proposed algorithm is demonstrated through some numerical experiments. The outline of this paper is organized as follows. First, in Section 2, we briefly review the concepts of possibility, probability, fuzzy variable, random variable and fuzzy random variable. Then we describe the assumptions and notations and one new types of fuzzy random scheduling model in Section 3. In Section 4, we give solution method about this model. And then, algorithm and application example are present in Section 5 and 6 respectively. Finally, we concludes this paper with a summary.

2 Preliminaries

2.1 Fuzzy random variable

Fuzzy random theory is an important part in mathematics field. In order to expand our study on it, we must clear about which are fuzzy random variables first of all.

Definition 1. *[17] Let Ω be a set of all outcomes of random experiment. A (nonempty) collection $\mathcal{A}$ of subset (called event) of Ω is assumed to have the following properties:*
(i) $\Omega \in \mathcal{A}$;
(ii) If $A \in \mathcal{A}$, then $A^c \in \mathcal{A}$; and
(iii) If $A_n \in \mathcal{A}$ is a countable sequence of events, the $\cup_n A_n \in \mathcal{A}$. Such a collection $\mathcal{A}$ is called $\sigma-$algebra. For each random event A there is a nonnegative number $\Pr\{A\}$, called its probability, such that
(i) $\Pr\{\varnothing\} = 0, \Pr\{\Omega\} = 1$; and
(ii) $\Pr\{\cup_n A_n\} = \sum_n \Pr\{A_n\}$ for every countable sequence of mutually disjoint events A_n. The triplet $(\Omega, \mathcal{A}, \Pr)$ is called a probability space, and the function $\Pr$ is referred to as a probability measure.

Definition 2. *[17] A random variable on the probability space $\Omega \in \mathcal{A}$ is a function ω from Ω to the real line $\Re$.*

Definition 3. *[17] Let Θ be a nonempty set experiment, and $\mathcal{P}(\Theta)$ be the power set of Θ. For each $A \in \mathcal{P}(\Theta)$, there is a nonnegative number $\mathrm{Pos}\{A\}$, called its possibility, such that*
(i) $\mathrm{Pos}\{\varnothing\} = 0, \mathrm{Pos}\{\Omega\} = 1$; and
(ii) $\mathrm{Pos}\{\cup_k A_k\} = \sup_n \mathrm{Pos}\{A_k\}$ for any arbitrary collection $\{A_k\}$ in $\mathcal{P}(\Theta)$. The triplet $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$ is called a possibility space, and the function Pos is referred to as a possibility measure.

Definition 4. *[17] A fuzzy variable is defined as a function from the possibility space $(\Theta, \mathcal{P}(\Theta), \mathrm{Pos})$ to the real line $\Re$.*

Definition 5. *[16] A fuzzy random variable is a function ξ from a probability space $(\Omega, \mathcal{A}, \Pr)$ to the set of fuzzy variable such that $Cr\{\xi(\omega) \in B\}$ is a measurable function of ω for and Borel set B of $\Re$.*

Example 1. let $(\Omega, \mathcal{A}, \Pr)$ be a probability space. If $\Omega = \omega_1, \omega_2, \cdots, \omega_m$ and $u_1, u_2, \cdots, u_m$ are fuzzy variable, the the function

$$\xi = \begin{cases} u_1 + \omega_1, & \text{if } \omega = \omega_1, \\ u_2 + \omega_2, & \text{if } \omega = \omega_2, \\ u_3 + \omega_3, & \text{if } \omega = \omega_3, \\ \vdots & \vdots \\ u_m + \omega_m, & \text{if } \omega = \omega_m, \end{cases}$$

is clearly a fuzzy random variable.

3 Machine scheduling problem

At first, we get the general conception of fuzzy random variable. Later, we put forward parallel machine scheduling models. Let us see following assumption and notation:

Assumption:

(i) each job has only one operation and can be processed on any machine;

(ii) each machine can process only one job at a time;

(iii) all jobs are available for machine processing simultaneously at time zero;

(iv) the times are assumed to be deterministic;

(v) the due times are assumed to be fuzzy random variable.

Notations:

$i = 1, 2, \cdots, n$: the jobs to be scheduled;

$k = 1, 2, \cdots, m$: the machines;

t_{ik}: the deterministic time of job i on machine k $(i = 1, 2, \cdots, n, k = 1, 2, \cdots, m)$;

$\widetilde{D}_i$: the fuzzy random due-date of job i $(i = 1, 2, \cdots, n)$;

(x, y): the decision vector.

In the vector, $x = (x_1, x_2, \cdots, x_n)$ integer decision variables representing n jobs with $1 \leq x_i \leq n$ and $x_i \neq x_j$ for all $i \neq j$, $(i, j = 1, 2, \cdots, n)$, $y = (y_1, y_2, \cdots, y_{m-1})$ integer decision variables with $y_0 \equiv\leq y_1 \leq y_2 \leq \cdots \leq y_{m-1} \leq n \equiv y_m$.

We note that the schedule is fully determined by the decision variables x and y in the following way. For each $k(1 \leq k \leq m)$, if $y_k \neq y_{k-1}$, then machine k is not used, if $y_k > y_k - 1$, then machine k is used and processes the following jobs in turn: $x_{y_{k-1}} + 1 \rightarrow x_{y_{k-1}} + 2 \rightarrow \cdots \rightarrow x_{y_k}$.

Let $C_i(x, y)$ be the completion times of jobs $i, i = (1, 2, \cdots, n)$, respectively. Then the completion times can be calculated by the following equations:

$$c_{x_{y_{k-1}}+1}(x, y) = t_{x_{y_{k-1}}+1k}, \tag{1}$$

and

$$c_{x_{y_{k-1}}+j}(x, y) = c_{x_{y_{k-1}}+j-1}(x, y, \xi) + t_{x_{y_{k-1}}+jk}, \tag{2}$$

for $2 \leq j \leq y_k - y_{k-1}$ and $k = (1, 2, \cdots, m)$.

According to job's completion times, satisfaction levels are assigned for them based on the membership function $\mu_{\widetilde{D_i}}(c_i)$, respectively.

We define the functions as follows:

$$\mu_{\widetilde{D_i}}(c_i) = \begin{cases} 1, & c_i < d_i, \\ W_i(c_i - d_i), & d_i < c_i < di + q_i, \\ 0, & di + q_i < c_i. \end{cases} \tag{3}$$

Here W_i is a strictly decreasing function satisfying $W_i(0)$=1, q_i denotes $sup\{x|W_i(x) > 0\}$ and d_i, which we call an expected due-date, is a standard normal distributed random variable. Then, the probability density function for d_i is

$$f_i(t) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}, & t \geq 0, \\ 0, & t < 0. \end{cases} \tag{4}$$

Define $W_i(x)$ a linear function as $-b_i x + 1$. For instance, the membership function is as in Fig. 1.

We consider the following problem:

$$\begin{cases} \min \quad \{E[\mu_{\widetilde{D_i}}(d_i)] \\ \text{subject to} \\ 1 \leq x_i \leq n_i, \quad i = 1, 2, \cdots, n, \\ x_i \neq x_j, i \neq j, \quad i, j = 1, 2, \cdots, n, \\ 0 \leq y_1 \leq y_2 \leq \cdots \leq y_{m-1} \leq n, \\ x_i, y_j, \quad i = 1, 2 \cdots n, \ j = 1, 2 \cdots m-1, \text{integers}. \end{cases} \tag{5}$$

Obviously, we can see $x_i, y_j, (i = 1, 2 \cdots n, j = 1, 2 \cdots m - 1)$ are decision variables, we don't refer tardy and idle.

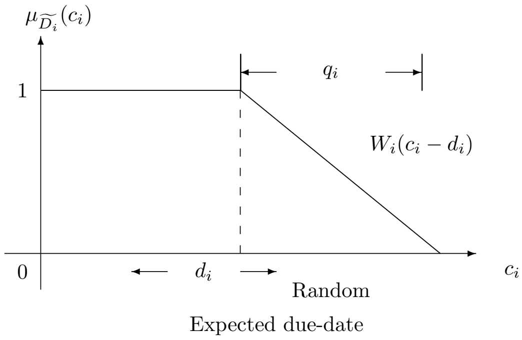

Fig. 1. Membership function for fuzzy random due-date

4 Solution method

As c_i is given, we must consider fluctuation parameter d_i, so the membership function can be written as follows:

$$\mu_{\widetilde{D_i}}(d_i) = \begin{cases} 1, & c_i < d_i, \\ W_i(c_i - d_i), & d_i < c_i < di + q_i, \\ 0, & di + q_i < c_i. \end{cases} \tag{6}$$

Then

$$E[\mu_{\widetilde{D_i}}(d_i)] = \int_0^{\infty} \mu_{\widetilde{D_i}}(t) f_i(t)dt = \int_{c_i - q_i}^{c_i} W_i(c_i - t) f_i(t)dt + \int_{c_i}^{\infty} f_i(t)dt. \tag{7}$$

As the density function is standard normal density function, we can calculate $E[\mu_{\widetilde{D_i}}(d_i)]$ in this way:

$$\begin{aligned} E[\mu_{\widetilde{D_i}}(d_i)] &= \int_{c_i - q_i}^{c_i} \{-b_i(c_i - t) + 1\} f_i(t)dt + \int_{c_i}^{\infty} f_i(t)dt \\ &= \int_{c_i - q_i}^{c_i} (-b_i c_i + 1) f_i(t)dt + b_i \int_{c_i - q_i}^{c_i} t f_i(t)dt + \int_{c_i}^{\infty} f_i(t)dt \\ &= (1 - b_i c_i) \int_{c_i - q_i}^{c_i} dF_i(t) + b_i \int_{c_i - q_i}^{c_i} t dF_i(t) + 1 - F_i(c_i) \\ &= (1 - b_i c_i)[F_i(c_i) - F_i(c_i - q_i)] + \frac{1}{2} b_i \int_{c_i - q_i}^{c_i} f_i(t) dt^2 + 1 - F_i(c_i) \\ &= -(b_i q_i - 1)\Phi(c_i - q_i) - b_i c_i \Phi(c_i) - b_i e^{-\frac{c_i^2}{2}} + b_i e^{-\frac{(c_i - q_i)^2}{2}} + 1. \end{aligned}$$

Set $q_i = \frac{1}{b_i}$, we get

$$E[\mu_{\widetilde{D_i}}(d_i)] = (1 - b_i^2)\Phi(c_i - b_i) - b_i c_i \Phi(c_i) - b_i e^{-\frac{c_i^2}{2}} + b_i e^{-\frac{(c_i - b_i)^2}{2}} + 1. \tag{8}$$

Now, we put a fuzzy stochastic problem into a new one as follows:

$$\begin{cases} \min \quad \{E[\mu_{\widetilde{D_i}}(d_i)] = (1 - b_i^2)\Phi(c_i - b_i) - b_i c_i \Phi(c_i) - b_i e^{-\frac{c_i^2}{2}} + b_i e^{-\frac{(c_i - b_i)^2}{2}} + 1 \\ \text{subject to} \\ 1 \le x_i \le n_i, \quad i = 1, 2, \cdots, n, \\ x_i \ne x_j, i \ne j, \quad i, j = 1, 2, \cdots, n, \\ 0 \le y_1 \le y_2 \le \cdots \le y_{m-1} \le n, \\ x_i, y_j, \quad i = 1, 2 \cdots n, \ j = 1, 2 \cdots m - 1, \text{integers}. \end{cases} \tag{9}$$

5 Algorithm

Genetic algorithm of this problem shows as follows:

(1) representation structure: We use a vector $x = (x_1, x_2, \ldots, x_n)$ as a chromosome to represent a solution to the optimization problem.

(2) handling the constraints: To ensure the chromosomes generated by genetic operators are feasible, we can use the technique of fuzzy random simulation to check them.

(3) initializing process: Suppose that the DM is able to predetermine a region which contains the feasible set. Generate a random vector x from this region until a feasible one is accepted as a chromosome. Repeat the above process $N_{pop-size}$ times, then we have $N_{pop-size}$ initial feasible chromosomes $x_1, x_2, \ldots, x^{N_{pop-ize}}$.

(4) evaluation function: The regret value of each chromosome x is calculated, then the fitness function of each chromosome is computed by

$$eval(x) = \frac{r^{max} + r(x,p) + \varepsilon}{r^{max} - r^{min} + \varepsilon},$$

where $\varepsilon \in (0,1)$, r^{max} and r^{min} denote the maximal regret value and minimum regret value in the current generation, respectively.

(5) selection process: The selection process is based on spinning the roulette wheel $N_{pop-size}$ times. Each time a single chromosome for a new population is selected in the following way: calculate the cumulative probability q_i for each chromosome x_i, generate a random number r in $[0, q^{N_{pop-size}}]$, and select the ith chromosome x^i such that $q_{i-1} < r < q_i$, $1 < i < N_{pop-size}$. Repeat the above process $N_{pop-size}$ times and we obtain $N_{pop-size}$ copies of chromosomes.

(6) crossover operation: Generate a random number c from the open interval (0, 1) and the chromosome x_i is selected as a parent provided that $c < P_c$, where parameter P_c is the probability of crossover operation. Repeat this process $N_{pop-size}$ times and $P_c \cdot N_{pop-size}$ chromosomes are expected to be selected as to undergo the crossover operation. The crossover operator on x^1 and x^2 will produce two children y^1 and y^2 as follows:

$$y^1 = cx^1 + (1-c)x^2; \quad y^2 = cx^2 + (1-c)x^1.$$

If both children are feasible, then we replace the parents with them, or else we keep the feasible one if it exists. Repeat the above operation until two feasible children are obtained or a given number of cycles is finished.

(7) mutation operation: Similar to the crossover process, the chromosome x_i is selected as a parent to undergo the mutation operation provided that random number $m < P_m$, where parameter Pm as the probability of mutation operation. $P_m \cdot N_{pop-size}$ chromosomes are expected to be selected after repeating the process $N_{pop-size}$ times. Suppose that x_1 is chosen as a parent. Choose a mutation direction $d \in \Re^n$ randomly. Replace x with $x + M \cdot d$ if $x + M \cdot d$ is feasible, otherwise we set M as a random number between 0 and M until it is feasible or a given number of cycles is finished. Here, M is a sufficiently large positive number. We illustrate genetic algorithm procedure as follows:

Step 1. Input the parameters $N_{pop-size}$, P_c and P_m.
Step 2. Initialize $N_{pop-size}$ chromosomes.
Step 3. Update the chromosomes by crossover and mutation operations.
Step 4. Compute the fitness of each chromosome based on the regret value.
Step 5. Select the chromosomes by spinning the roulette wheel.
Step 5. Repeat the second to fourth steps for a given number of cycles.
Step 6. Report the best chromosome as the optimal solution.

6 An application example

We take a small manufactory as an example.

In a factory, we have three machines and fifteen jobs. In order to advance the efficiency, further more production value, we must schedule the job on which machine and machines process jobs in which sequence considering the due-dates and processing times.

Although each machine can do the same work, because of used years or normal or abnormal operation, the function has difference. In the other word, it means the same job on the different machine may process distinctive time. According to previous work experience, the processing times for each job are deterministic in this factory. We list the processing times in Table 1.

Table 1. Processing times

jobs	Processing times on machine 1	Processing times on machine 2	Processing times on machine 3	jobs	Processing times on machine 1	Processing times on machine 2	Processing times on machine 3
1	10	11	10	9	13	12	13
2	8	9	10	10	2	3	2
3	7	7	6	11	10	11	10
4	8	10	11	12	8	9	10
5	2	4	3	13	7	7	6
6	7	5	6	14	8	10	11
7	5	6	8	15	2	4	3
8	12	9	10				

Every job has its due-dates.In this factory, the administrators take it as an uncertain one. Their satisfaction levels are assigned for them based on the membership function $\mu_{\widetilde{D_i}}(c_i)$, respectively.

We define the functions as follows:

$$\mu_{\widetilde{D_i}}(c_i) = \begin{cases} 1, & c_i < d_i, \\ W_i(c_i - d_i), & d_i < c_i < di + q_i, \\ 0, & di + q_i < c_i. \end{cases} \tag{10}$$

Here W_i is a strictly decreasing function satisfying $W_i(0)$=1, q_i denotes $sup\{x|W_i(x) > 0\}$ and d_i is a standard normal distributed random variable. Then, the probability density function for d_i is

$$f_i(t) = \begin{cases} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}, & t \geq 0, \\ 0, & t < 0. \end{cases} \tag{11}$$

Define $W_i(x)$ a linear function as $-b_i x + 1$.

All of these are the same as in the Section 3. So the engineers who schedule the machines and jobs can utilize the method in our paper to get the satisfactory solution.

They has coded with Matlab and run on PC. The parameters are set as follows: the pop-size is 500, the probability of crossover P_c is 0.7, the probability of mutation P_m is 0.06.

And basing on the practical situation, they set

$$\begin{aligned} &(b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}) \\ &\quad = (0.01, 0.2, 0.03, 0.02, 0.1, 0.04, 0.02, 0.06, 0.1, 0.04, 0.05, 0.2, 0.03, 0.08, 0.3, 0.02) \end{aligned}$$

A run of the genetic algorithm (400 generations in GA) shows the optimal scheduling is

Machine 1: $11 \longrightarrow 8 \longrightarrow 10 \longrightarrow 5 \longrightarrow 13$

Machine 2: $7 \longrightarrow 4 \longrightarrow 2 \longrightarrow 15 \longrightarrow 12 \longrightarrow 1 \longrightarrow 6$

Machine 3: $9 \longrightarrow 14 \longrightarrow 3$.

7 Conclusions

With the development of economy and especially in recent competitive period with tiny profit, manufacturing corporation entrepreneurs demand most ideal outcome. Operational research is extensively used for optimum models, and then we get the result, providing feasible schemes to management experts.

In this paper, we have demonstrated one type of fuzzy random scheduling models on machines and give the algorithm. In our further study, We may give out an effective genetic algorithm to solve the proposed models and provide efficiently computational studies. It may be concluded that the modeling methods can be extended to other fuzzy random scheduling problems.

References

[1] V. A. Armentano, D. P. Ronconi. Tabu search for total tardiness minimization in flow shop scheduling problems. *Computers and Operations Research*, 1999, **26**: 219-235.

[2] D. Dubois, H. Fargier, H. Prade. Fuzzy constraints in job-shop scheduling. *Journal of Intelligent Manufacturing*, 1995, **6**: 215-234.

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

[4] H. Ishibuchi, N. Yamamoto, T. Murata, et al. Genetic algorithms and neighborhood search algorithms for fuzzy flow shop scheduling problems. *Fuzzy Sets and Systems*, 1994, **67**: 81-10.

[5] H. Ishii, M. Tada, T. Masuda. Two scheduling problems with fuzzy due-dates. *Fuzzy Sets and Systems*, 1992, **46**: 339-347.

[6] T. Itoh, H. Ishii. Fuzzy due-date scheduling problem with fuzzy processing time. *International Transactions in Operational Research*, 1999, **6**: 639-647.

[7] T. Itoh, H. Ishii. One machine scheduling witt fuzzy random due-dates. *Fuzzy Optimal and Decision Making*, 2005, **4**: 71-78.

[8] T. Konno, H. Ishii. An open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint. *Fuzzy Sets and Systems*, 2000, **109**: 141-147.

[9] M. Kuroda, Z. Wang. Fuzzy job shop scheduling. *International Journal of Production Economics*, 1996, **44**: 45-51.

[10] M. Litoiu, R. Tadei. Real-time task scheduling with fuzzy deadlines and processing times. *Fuzzy Sets and Systems*, 2001, **117**: 35-45.

[11] M. K. Luhandjula. Fuzziness and randomness in optimization framework. *Fuzzy Sets and System*, 1996, **77**: 291-297.

[12] M. K. Luhandjula, H. lchihashi, M. Inuiguchi. Fuzzy and semi-infinite optimization. *Information Science*, 1992, **61**: 233-250.

[13] B. Liu. Dependent-chance programming in fuzzy environments. *Fuzzy Sets and Systems*, 2000, **109**(1): 97-106.

[14] B. Liu. Dependent-chance programming with fuzzy decisions. *IEEE Transactions on Fuzzy Systems*, 1999, **7**(3): 354-360.

[15] Y. K. Liu, B. Liu. A class of fuzzy random optimization: Expected value models. *Information Science*, 2003, **155**(1): 89-102.

[16] Y. K. Liu, B. Liu. Fuzzy random variables: A scalar expected value operator. *Fuzzy Optimization and Desicion Making*, 2003, **2**: 143-160.

[17] B. Liu. *Theory and Practice of Uncertain Programming*. Physica-Verlag, New York, 2002.

[18] C. S. McCahon, E. S. Lee. Job sequencing with fuzzy processing times. *Computers and Mathematics with Applications*, 1990, **19**: 31-41.

[19] H. Prade. Using fuzzy set theory in a scheduling problem: A case study. *Fuzzy Sets and Systems*, 1979, **2**: 153-165.

[20] T. P. Hong, C. M. Huang, K. M. Yu. LPT scheduling for fuzzy tasks. *Fuzzy Sets and Systems*, 1998, **97**: 277-286.