Fuzzy Programming for Parallel Machines Scheduling: Minimizing Weighted Tardiness/Earliness and Flow time through Genetic Algorithm

Mohammad Asghari[a,]*, Samaneh Nezhadali[b]

[a] MSc, Department of industrial engineering, Ferdowsi University of Mashhad, Azadi Sq., Mashhad, Iran
[b] MSc, Department of management, Iran Chamber of Commerce, Industries and Mines, Mashhad, Iran
Received 2 June, 2013; Revised 20 February, 2014; Accepted 08 March, 2014

Abstract

Appropriate scheduling and sequencing of tasks on machines is one of the basic and significant problems that a shop or a factory manager encounters; this is why in recent decades extensive researches have been done on scheduling issues. A type of scheduling problems is just-in-time (JIT) scheduling and in this area, motivated by JIT manufacturing, this study investigates a mathematical model for appraising a multi-objective programing that minimize total weighted tardiness, earliness and total flow time with fuzzy parameters on parallel machines, simultaneously with respect to the impact of machine deterioration. Besides, in this paper, attempts are made to present a defuzzification approach and a heuristic method based genetic algorithm (GA) to solve the proposed model. Finally, several dominance properties of optimal solutions are demonstrated in comparison with the results of a state-of-the-art commercial solver and the simulated annealing method that is followed by illustrating some instances for indicating validity and efficiency of the method.
Keywords: Mathematical optimization, Fuzzy multi-objective model, Parallel machines scheduling, Weighted tardiness/earliness, Genetic algorithm.

1. Introduction

In classical scheduling problems, the processing time of job has been assumed constant. However, there are many situations that this time may be subject to change due to deterioration and/or learning phenomena (McKay et al. 2002).

Scheduling with costs of earliness and tardiness has received considerable and increasing attention in recent researches. In many practical situations, it is required to guarantee that as many jobs as possible to meet their due dates (i.e., to minimize the number of tardy jobs) since in such cases, having a job missing its due date is very costly. Thus, minimization of the number of tardy jobs should be the primary concern. On the other hand, it is desirable to minimize the job earliness to minimize the inventory cost. Early/tardy scheduling problems are compatible with the concepts of just-in-time production and supply chain management, which have been adopted by many organizations. Indeed, these production strategies view both early and tardy deliveries as undesirable. By the machine deterioration effect, we mean that each machine deteriorates at a different rate. This deterioration is considered in terms of cost that depends on the production rate, the machine operating characteristics and the kind of work done by each machine. Moreover, job-processing times are increasing functions of their starting times and follow a simple linear deterioration. Browne and Yechiali (1990) first introduced it. Since then, deteriorating job scheduling problems have been widely discussed. Ruat et al. (2008) considered the problem of scheduling a given number of jobs on a single machine with time deteriorating job values and capacity constraints while the objective function is to maximize total revenue. Gawiejnowicz et al. (2006) considered a single machine time-dependent scheduling problem. They introduced two scenarios for a given sequence of job deterioration and formulated a greedy polynomial time approximation algorithm for each scenario.

In recent decades, most of the researches have focused on Just-In-Time (JIT) scheduling models. For example, Sridharan and Zhou (1996) considered a single machine problem with total weighted earliness and tardiness and developed a solution procedure based on decision theory. Cai and Zhou (1999) studied a parallel machine stochastic scheduling problem to minimize expected total cost for

* Corresponding author Email address: hooman.asghari@outlook.com

objective functions in details. Next, the mathematical formulation is developed for the problem. In Section 3, we present a new solution method for it and describ an approach in order to consider the four objectives as a single objective. In Section 4, some numerical examples of its occurrence are applied and the feasibility and effectiveness of the proposed method are demonstrated by comparing with the simulated annealing method. Finally, concluding remarks are given in the last Section.

2. Problem Formulation

The following notations and definitions are used to describe a multi-objective on parallel machines scheduling problem that is an extension of studied problem by Mazdeh et al. (2010).

This problem considers a set of N independent jobs, $J_1, J_2, . . . , J_n$, on a number of parallel machines selected from a set of M potential machines as each of these jobs exactly need one operation on one machine. Each job J_i has a processing time $\tilde{p}_j$ and a due date $\tilde{d}_j$ that all processing times and due dates are considered as fuzzy numbers. Here machines are supposed to become worse at a different rate by allocating and then doing the jobs on them. This deterioration is a function of production rate, machine's operating characteristics and the kind of work accomplished by each machine, which considered in terms of cost.

Sets

N	The set of jobs that must be scheduled
M	The set of available machines

$i,j \in \{0,1,...,N\}$ are designated job, where job 0 is a dummy job and is always at the first position on a machine

Parameters

γ_i	Earliness weight of job i
β_i	Tardiness penalty of job i
r_i	Arrived time of job i to queue
$\tilde{d}_i$	Due date of job i
$\tilde{a}_{im}$	Processing fix time of job i on machine m
$\tilde{b}_i$	Growth rate of the processing time of job i
$\tilde{c}_{im}$	Cost of machine deterioration

Decision variables

X_{ijm}	1, if job j immediately follows job i in sequence on machine m; 0, otherwise
Y_{im}	1, if job i assigned to machine m; 0, otherwise
$\tilde{T}_i$	Tardiness value of job i
$\tilde{E}_i$	Earliness value of job i
$\tilde{P}_{im}$	Processing time of job i on machine m
$\tilde{S}_{im}$	Starting time of job i on machine m
$\tilde{C}_i$	Completion time of job i

A job is early if its completion time is smaller than the common due date. On the other hand a job is tardy if its processing ends after due date. It is not known in advance whether a job will be completed before or after the due date.

The notations and other assumptions applied in mathematical formulation are as follows.

2.1. Problem assumptions

The following notations are the assumptions considered in the present model.

- Each machine is able to process each job;
- The machine can process at most one job at a time;
- No processing is allowed;
- Associated with job j ($j=1, \ldots ,n$) there are a processing time $\tilde{p}_j$ and a due date $\tilde{d}_j$;
- Job processing time may be different by various machines;
- Job processing time is described by a function of the starting time ($\tilde{P}_{jm} = a_{jm} + \tilde{b}_j\tilde{S}_{jm}$);
- The growth rate of the processing time ($\tilde{b}_j$) is independent of machine;
- The jobs are considered independent of each other;

2.2. Sets and indices

The following shows nomenclature used in the model.

2.3. Mathematical model

Base on the aforementioned descriptions and indices, a fuzzy nonlinear programming model is developed as follows:

Minimize

$$F1 = \sum_{i \in N} \gamma_i \tilde{E}_i \quad (1)$$

$$F2 = \sum_{i \in N} \beta_i \tilde{T}_i \quad (2)$$

$$F3 = \sum_{i \in N} \tilde{C}_i - r_i \quad (3)$$

$$F4 = \sum_{i \in N} \sum_{m \in M} Y_{im} . \tilde{c}_{im} \quad (4)$$

Subject to

$$\sum_{i \in N} X_{0im} \leq 1 \qquad \forall m \in M, \quad (5)$$

$$\sum_{i \in N, i \neq j} \sum_{m \in M} X_{ijm} = 1 \qquad \forall j \in N, \quad (6)$$

$$\sum_{j \in N, i \neq j} X_{ijm} \leq Y_{im} \qquad \forall i \in N, \forall m \in M, \quad (7)$$

$$\sum_{i \in N, i \neq j} X_{ijm} \leq Y_{jm} \qquad \forall j \in N, \forall m \in M, \quad (8)$$

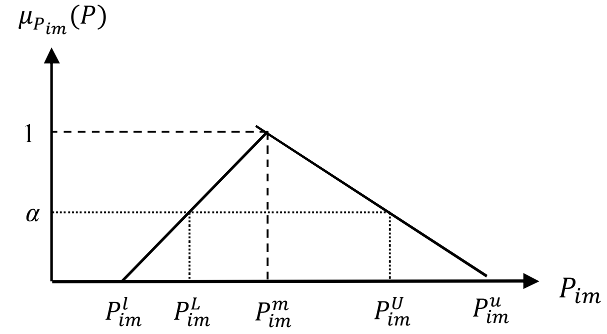

Fig. 2. α-cut on membership function of triangular fuzzy number $\tilde{P}_{im}$

According to Fig 2 following equations can be resulted:

$$\frac{\alpha}{P_{im}^L - P_{im}^l} = \frac{1}{P_{im}^m - P_{im}^l} \Rightarrow \alpha\left(P_{im}^m - P_{im}^l\right) = P_{im}^L - P_{im}^l \Rightarrow P_{im}^L = \alpha . P_{im}^m + (1-\alpha)P_{im}^l \quad (20)$$

$$\frac{\alpha}{P_{im}^u - P_{im}^U} = \frac{1}{P_{im}^u - P_{im}^m} \Rightarrow \alpha(P_{im}^u - P_{im}^m) = P_{im}^u - P_{im}^U \Rightarrow P_{im}^U = \alpha . P_{im}^m + (1-\alpha)P_{im}^u \quad (21)$$

Others like $\tilde{P}_{im}$ are converted into the interval form through α-cut on these numbers. However, due date ($\tilde{d}_i$) defined different with membership function as following that has been introduced same Hwang and Yoon (1981) and has been illustrated in Fig 3.

$$\mu_{\tilde{d}_i}(d) = \begin{cases} 1, & d \le d_i^* \\ \dfrac{d - d_i^m}{d_i^* - d_i^m}, & d_i^* \le d \le d_i^m \\ 0, & d \ge d_i^m \end{cases} \quad (22)$$

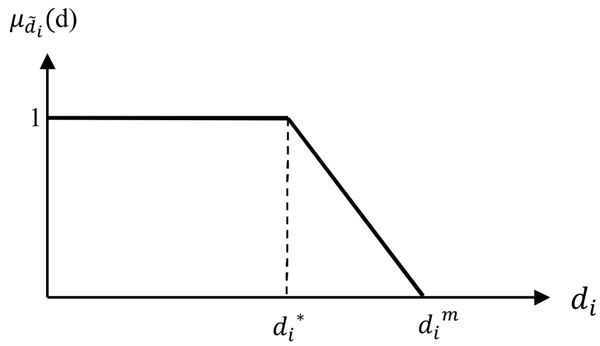

Fig. 3. Membership function of due date

This number can be converted to the interval form by applying α-cut as follows:

$$\tilde{d}_{i_\alpha} = [d_i^L , d_i^U] = [0 \ , \ \alpha . d_i^* + (1-\alpha)d_i^m] \quad \text{Interval number (2)}$$

3.2. Conversion to deterministic programming

If we substitute interval numbers in the model, fuzzy programming is converted to interval programming. Now, the numbers should be defuzzified. In this paper, deterministic numbers are obtained via applying convex conversion. Hence, the interval programming converts into deterministic programming.

Follows, interval numbers convert into deterministic form by applying convex conversion context that present in Appendix A.

3.3. Solving multi-objective optimization

I use a general form of multi-objective programming that is a family of L_p-metrics and is adopted from Hwang & Yoon, (1981). This method considers the minimum deviation from the ideal solution as follows:

$$\begin{aligned} &\text{Min} \ \ f_1(x), f_2(x), \ldots, f_n(x) \\ &\text{S.t:} \ \ \ x \in X \end{aligned} \quad (23)$$

That $f_1(x), f_2(x), \ldots, f_n(x)$ are the objective functions and x is the feasible region. First, an ideal solution for each objective function separately will be obtained by following problems solving:

$$\begin{aligned} &f_i^* = \text{Min} \, f_i(x) \qquad (i=1, \ldots, n) \\ &\text{S.t:} \ \ x \in X \end{aligned} \quad (24)$$

Then, will be obtained without unit function with dividing the each function in its optimum value. Thus, multi-objective programming problem can satisfactorily solve by following new objective function:

$$\begin{aligned} &\text{Min} \left[\sum_i \omega_i^p \left(\frac{f_i(x) - f_i^-}{f_i^+ - f_i^-} \right)^p \right]^{\frac{1}{p}} \\ &\text{S.t:} \qquad x \in X \end{aligned} \quad (25)$$

Where each function is weighted using ''ω'' to denote the importance of objective functions. This weight adjustment is used for alimenting and balancing between functions that will be determined by decision makers just as following relationship can be established.

$$\begin{aligned} &\sum_i \omega_i = 1 \\ &\omega_i \ge 0 \qquad i \in (1, \ldots, n) \end{aligned} \quad (26)$$

Obviously, the result is dependent on the value of p. Generally, p is 1 or 2. However, other values of p also can be used.

4.2. Simulated annealing

In this paper, for comparison, simulated annealing (SA) (Davis 1987; Kirkpatrick et al. 1983) is adopted as an another search method for the problem. Here, observe that SA searches for solutions by exchanging the job processing order for each machine.

The algorithm of SA used in this paper is summarized as follows.

Step 1. Generate one solution (schedule) through the random selection in Step 4 of an active scheduling generating algorithm and denote it by X^c. Set an initial temperature T_0.

Step 2. Represent the job process sequence for each machine of a solution X^c by the corresponding matrix, and select one machine at random. Select two jobs of the machine and exchange them. For example in the problem of 3 jobs and 3 machines, when the first job (J_3) and the second job (J_1) of machine 3 (M_3) are selected and the result after exchange becomes as shown in Fig 5.

$$\begin{matrix} M_1 \\ M_2 \\ M_3 \end{matrix} \begin{pmatrix} J_1 & J_2 & J_3 \\ J_2 & J_1 & J_3 \\ \boxed{J_3} & \boxed{J_1} & J_2 \end{pmatrix} \Rightarrow \begin{matrix} M_1 \\ M_2 \\ M_3 \end{matrix} \begin{pmatrix} J_1 & J_2 & J_3 \\ J_2 & J_1 & J_3 \\ \boxed{J_1} & \boxed{J_3} & J_2 \end{pmatrix}$$

Fig. 5. Example of job processing order and job exchange

Step 3. Based on the job processing sequence after job exchange, dissolve the conflict occurred in Step 4 of an active scheduling generating algorithm, and generate a new solution. If the obtained solution is different from the solution before job exchange, set the solution as a neighborhood solution X and go to Step 4. Otherwise, return to Step 2, and select a new exchange pair.

Step 4. If the objective function value of the solution through exchange is improved, accept the exchange, and set $X^c = X$. Otherwise, determine the acceptance by the following substeps.

1. Using the decrement Δf of the objective function value and temperature T, calculate exp ($-\Delta f / T$).
2. Generate a uniform random number on the open interval (0, 1) and compare it with the value of exp ($-\Delta f / T$).
3. If the value of exp ($-\Delta f / T$) is greater than the random number, accept the exchange, and set $X^c = X$. Otherwise, the exchange is not accepted.

When the exchange is accepted, go to Step 5. Otherwise, return to Step 2 to find the next exchange pair.

Step 5. The equilibrium state test is performed by checking whether the change of the objective function value obtained through the exchanges in the prescribed number of times is small enough or not. The number for the equilibrium state test is called the epoch. Here, the test is performed in the following substeps.

1. Repeat the procedures from Step 2 to Step 4, until the exchanges are performed by the number of epoch. When reached the epoch number, perform the following substeps (2)-(4).
2. Calculate the average value $\bar{f}_e$ of the objective function values during the current epoch and the average value $\bar{f}'_e$ of the objective function values through the exchanges thus far.
3. Check whether the relative error between the average value $\bar{f}'_e$ in the whole and the average value $\bar{f}_e$ during the epoch is smaller than the prescribed tolerance value ε or not, i.e., check whether $\left(\left|\bar{f}_e - \bar{f}'_e\right| / \bar{f}'_e\right) < \varepsilon$ holds or not.
4. When the relative error is smaller than the tolerance value, regard the equilibrium state is reached at this temperature, and go to Step 6 to decrease the temperature. Otherwise, clear the counter of the epoch, and return to Step 2 to repeat the job exchange process.

Step 6. Starting with an initial temperature T_0, decrease the temperature with the predetermined ratio α, i.e., $T_{\text{new}} = \alpha \times T_{\text{old}}$.

Step 7. If the number of the pair exchanges reaches the predetermined number, stop the algorithm.

Repeating this process, when the algorithm is terminated, select the solution with the best objective function value among the obtained solutions.

5. Numerical Example

According to our observations, there is no comparable mathematical model in the literature to compare with the proposed model. So, first a small test problem has been solved only to investigate behavior of proposed mathematical model. Tables 1, 2 and 3 summarize the data used for two numerical examples with 10 jobs.

Here, the behavior of the proposed fuzzy model is appraised for different α (α ϵ [0, 1]) through two solving methods. In Table 4, the model has been solved by the Lingo 13.0 solver and its computational times compared with the GA. The experiments were run in an Intel(R) core(TM) i3 CPU, at 2.13GHz and with 4.00 GB of RAM memory.

To solve the example problem, gave the common data of due dates, fixed part of the processing times and deteriorating cost. In our experiments, the population size is 30 and the new individuals are created with a crossover rate of 0.45. The termination criterion is the completion of 30 generations of individuals. The problem is solved for $p = 1$ and the importance weight of objective functions are determined as α_1=0.27, α_2=0.23, α_3=0.29 and α_4=0.21.

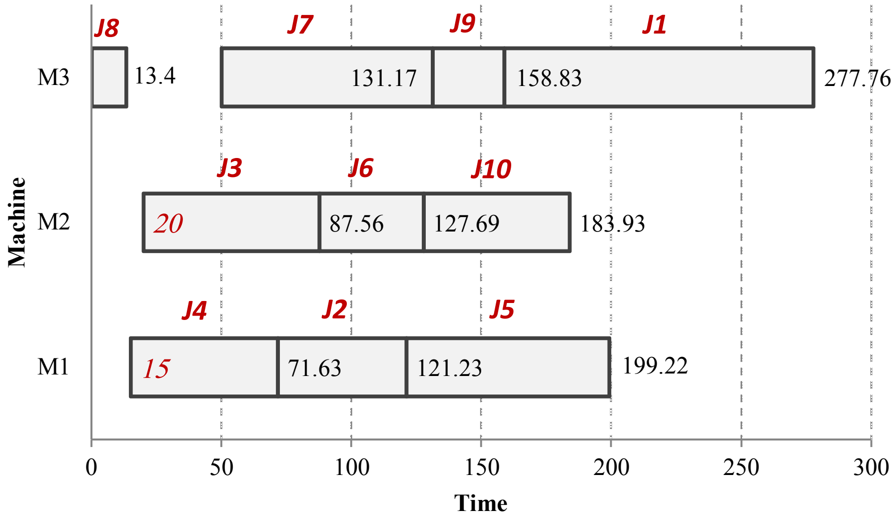

Fig. 6. Gantt chart for α = 0.2

Table 5
Test problems' dimensions.

Problem number		1	2	3	4	5	6	7	8	9	10	11	12	13	14
The number of jobs	N	10	15	20	25	30	40	50	70	100	150	200	300	400	500
The number of machines	M	4	5	10	10	15	20	25	30	40	50	70	100	150	200

Table 6
The values of the parameters used in the test problems

Parameter	Symbol	Range
Earliness weight	γ_i	~ uni[1,3]*
Tardiness penalty	β_i	~ uni[1,3]
Processing fix time	$\tilde{a}_{im}$	~ uni[0,60]
Processing variable time	$\tilde{b}_i$	~ uni[0,1]
Arrived time	r_i	~ uni[0,100]
Due date	$\tilde{d}_i$	~ uni[100.200]
Deteriorating cost	$\tilde{c}_{im}$	~ uni[0,15]

* Uniform distribution [lower bound, upper bound]

In Table 7, the evaluation results have been summarized through different α values (α ϵ {0.2, 0.5 and 1}), according to a group of parameters defined in Table 6. The parameter values of SA are set as 0.5, 0.9, 10'000, 5 and 0.2 that denote the initial temperature (T_0), the changing ratio (α), the number of search (S), the number of epoch and the tolerance value (ε), respectively. Here, an initial temperature is set to be 277 for the last instance when solving the problems which minimize the objective function using SA.

It should be emphasized here that these parameter values are found through a lot of experiences and these values are used in all of the trials of GA and SA. All of the trials of GA and SA are performed 10 times for each of the problems. The average times required for computation are respectively shown in Table 7.

and the results were compared by two heuristic methods, genetic algorithm and simulating annulling, for different illustrative examples to analyze and validate the approach. Computational results confirmed efficiency and effectiveness of the developed heuristic solution methods when time complexity is addressed.

Recent research has raised several issues that could be further investigated. For example, the accuracy and efficiency of the proposed method could be improved. A number of verification and validation methods may be helpful in testing the accuracy and consistency of the process.

References

Anglani A, Grieco A, Guerriero E, Musmanno R (2005) Robust scheduling of parallel machines with sequence-dependent set-up costs. European Journal of Operational Research 161:704-720.

Balin S (2011) Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation. Information Sciences 181:3551-3569.

Bilge Ü, Kıraç F, Kurtulan M, Pekgün P (2004) A tabu search algorithm for parallel machine total tardiness problem. Computers & Operations Research 31:397-414.

Browne S, Yechiali U (1990) Scheduling Deteriorating Jobs on a Single Processor. Operations Research 38:495-498.

Cai X, Zhou S (1999) Stochastic scheduling on parallel machines subject to random breakdowns to minimize expected costs for earliness and tardy jobs. Operations Research 47:422-437.

Cao D, Chen M, Wan G (2005) Parallel machine selection and job scheduling to minimize machine cost and job tardiness. Computers & Operations Research 32:1995-2012.

Davis L (1987) Genetic Algorithms and Simulated Annealing. Morgan Kaufmann (ed), Los Altos, CA.

Falkenauer E (1999) The worth of uniform crossover. In: Proceedings of the 1999 Congress on Evolutionary Computation CEC99, USA.

Gawiejnowicz S, Kurc W, Pankowska L (2006) Analysis of a time-dependent scheduling problem by signatures of deterioration rate sequences. Discrete Applied Mathematics 154:2150-2166.

Guner E, Erol S, Tani K (1998) One machine scheduling to minimize the maximum earliness with minimum number of tardy jobs. International Journal of Production Economics, 55:213-219.

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

Holland JH (1975) Adaptation in Natural and Artificial Systems. Arbor A (ed) The University of Michigan Press, MI.

Hong TP, Chuang TN (1999) A new triangular fuzzy Johnson algorithm. Computers & Industrial Engineering 36:179-200.

Hwang CL, Yoon K (1981) Multiple attribute decision making: Methods and applications. Springer-Verlag, Berlin and New York.

Ishibuchi H, Murata T (2000) Flow shop scheduling with fuzzy due date and fuzzy processing time. In: Slowinski R, Hapke M (ed) Scheduling under Fuzziness.

Ishii H, Tada M, (1995) Single machine scheduling problem with fuzzy precedence relation. European Journal of Operational Research 87:284-288.

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

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671-680.

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

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

Leu YY, Matheson LA, Rees LP (1994) Assembly line balancing using genetic algorithms with heuristic generated initial populations and multiple criteria. Decision Sciences 15:581–606.

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

Mazdeh MM, Zaerpour F, Zareei A, Hajinezhad A (2010) Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs. Applied Mathematical Modeling 34:1498-1510.

Mazzini R, Armentano VA (2001) A heuristic for single machine scheduling with early and tardy costs. European Journal of Operational Research 128:129-146.

McKay K, Pinedo M, Webster S (2002) Practice-focused research issues for scheduling systems. Production and Operations Management 11:249-258.

Oguz C, Ercan MF (2005) A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. J Scheduling 8:323-351.

Page J, Poli P, Langdon WB (1999) Smooth uniform crossover with smooth point mutation in genetic programming: a preliminary study. In: Genetic Programming, Proceedings of EuroGP'99, Sweden.

Peng J, Liu B (2004) Parallel machine scheduling models with fuzzy processing times. Information Sciences 166:49-66.

Pfund M, Fowler JW, Gadkari A, Chen Y (2008) Scheduling jobs on parallel machines with setup times and ready times. Computers & Industrial Engineering 54:764-782.

Piersma N, Romeijn HE (1996) Parallel machine scheduling: A probabilistic analysis. Naval Research Logistics (NRL) 43:897-916.

Prade H (1979) Using fuzzy set theory in a scheduling problem: A case study. Fuzzy Sets and Systems 2:153-165.

Radhakrishnan S, Ventura JA (2000) Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent set-up times. International Journal of Production Research 38:2233-2252.

Raut S, Gupta JND, Swami S (2008) Single machine scheduling with time deteriorating job values. Journal of the Operational Research Society 59:105-118.

Sridharan V, Zhou Z (1996) A decision theory based scheduling procedure for single machine weighted earliness and tardiness problems. European Journal of Operational Research 94:292-301.

Syswerda G (1989) Uniform crossover in genetic algorithms In: Schaffer DJ (ed) Proceedings of the 3rd International Conference on Genetic Algorithms, USA, pp. 2-9.

Wan G, Yen BPC (2002) Tabu search for single machine scheduling with distinct due windows and weighted