International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects

Oğuzhan Ahmet Arık & M. Duran Toksarı

International Journal of Production Research, 2018
Vol. 56, No. 7, 2488–2505, https://doi.org/10.1080/00207543.2017.1388932

Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects

Oğuzhan Ahmet Arık[a]* and M. Duran Toksarı[b]

[a]*Engineering and Architecture Faculty, Industrial Engineering Department, Istanbul Arel University, Istanbul, Turkey;* [b]*Engineering Faculty, Industrial Engineering Department, Erciyes University, Kayseri, Turkey*

(*Received 9 May 2017; accepted 26 September 2017*)

This paper investigates a multi-objective parallel machine scheduling problem under fully fuzzy environment with fuzzy job deterioration effect, fuzzy learning effect and fuzzy processing times. Due dates are decision variables for the problem and objective functions are to minimise total tardiness penalty cost, to minimise earliness penalty cost and to minimise cost of setting due dates. Due date assignment problems are significant for Just-in-Time (JIT) thought. A JIT company may want to have optimum schedule by minimising cost combination of earliness, tardiness and setting due dates. In this paper, we compare different approaches for modelling fuzzy mathematical programming models with a local search algorithm based on expected values of fuzzy parameters such as job deterioration effect, learning effect and processing times.

Keywords: parallel machines; fuzzy deterioration; fuzzy learning; due date assignment; local search

1. Introduction

With a practical view for scheduling problems, decision makers (DM) may not always determine exact scheduling parameters such as processing times and due dates because of lack of knowledge, DM's experience and judgement, vagueness and imprecision in scheduling environment for measuring parameters and uncertainty of problem's own characteristics. Processing times and due dates are usually exact and well-defined or deterministic in scheduling literature. However, there is a growing interest for scheduling problems with fuzzy parameters. For the readers, recent surveys (Abdullah and Abdolrazzagh-Nezhad 2014; Behnamian 2016) about scheduling problems with fuzziness may be guidance. First study about scheduling problems with fuzzy set theory was conducted by Prade (1979). Processing times in scheduling problems were firstly considered as fuzzy numbers by McCahon and Lee (1990). Ishii, Tada, and Masuda (1992) studied fuzzy due dates in scheduling problems for the first time. Some of other pioneer studies about fuzziness in scheduling problems were conducted by Tsujimura et al. (1993), Ishibuchi et al. (1994), Han, Ishii, and Fujii (1994), Dubois, Fargier, and Prade (1995), Ishii and Tada (1995) and Kuroda and Wang (1996). In scheduling problems with fuzziness, fuzzy parameters such as processing times (Chanas and Kasperski 2003; Niu, Jiao, and Gu 2008; Sakawa and Kubota 2000; Sakawa and Mori 1999; Yeh et al. 2014), due dates (Han, Ishii, and Fujii 1994; Ishibuchi et al. 1994; Ishii, Tada, and Masuda 1992; Li, Ishii, and Chen 2015), precedence relations (Li, Ishii, and Chen 2012, 2015; Xie, Xie, and Huang 2005) and batch sizes (Li, Ishii, and Masuda 2012) are studied together or individually.

Besides uncertainty of scheduling parameters such as processing times and due dates, some effects such as job deterioration and learning effects may change the initially measured processing times. The expression of job deterioration may be dependent on position number of the current job in the sequence or completion time of previous job. Job deterioration in a scheduling problem has time-increasing effect on processing times while jobs are waiting in the queue or being processed on the machines. Therefore, job deterioration effect can be called as a negative outcome for DM. The expression of learning effect implies that continuous repetition of similar jobs leads workers or the system to make the task being processed faster than previous. Each repetition of similar jobs on a machine or a processer has a time-decreasing effect on processing time. In other words, while job deterioration effect is a negative outcome, learning effect is a positive outcome for a scheduling problem. As a pioneer Biskup (1999) introduced learning effect for scheduling problem. Mosheiov (2001) was the first one who studied learning effect on parallel machine environment and also Mosheiov (2001) investigated several well-known objectives as one and multi-criteria on single machine problems under

*Corresponding author. Email: oguzhanarik@arel.edu.tr

learning effect. Some of significant reserches about scheduling problem under only learning effect were conducted by Kuo and Yang (2006), Biskup (2008), Mosheiov and Sidney (2003), Wang et al. (2008), Bachman and Janiak (2004), Mosheiov (2011), Koulamas and Kyparisis (2007), Yin et al. (2009), Cheng, Wu, and Lee (2008a), Wang and Xia (2005), Lee and Wu (2004), Janiak and Rudek (2009), Eren (2009), Qian and Steiner (2013a, 2013b) Rudek (2013) and Wang, Ng, and Cheng (2008).

Scheduling problems under only deterioration effect have been widely interested by researchers. As far as we know, Gupta and Gupta (1988) and Browne and Yechiali (1990) made first studies about deterioration effect. Mosheiov (1991) showed the optimal schedule is V-shaped for single machine scheduling problem when objective function is to minimise flow time. Mosheiov (1998) studied a scheduling problem with linear deterioration effect on identical parallel machines and proved the problem is NP-Hard. Some of other researches about scheduling problems under only deterioration effect were studied by Ng et al. (2002), Bachman, Janiak, and Kovalyov (2002), Mosheiov (2002), Wang and Xia (2006), Mosheiov (1995), Wu and Lee (2008), Wang et al. (2006), Wang, Ng, and Cheng (2008), Gawiejnowicz (2007) and Guo, Cheng, and Wang (2015).

Although, most of researches in the literature investigated these effects one by one in scheduling problems, there are lots of papers investigated both effects simultaneously. Wang (2006) was the first researcher who studied scheduling problems under both effects as far as we know. Wang (2007) investigated these effects on single machine environment and showed that the makespan, the sum of completion times, and the sum of completion times square minimisation problems all can be optimally solved by the SPT rule. Wang, Lin, and Shan (2008) considered a flow shop scheduling problem under learning and deterioration effects simultaneously and showed that for some special cases of the flow shop scheduling, the makespan minimisation problem and the total completion minimisation problem can be solved in polynomial time. Cheng, Wu, and Lee (2008b) investigated single machine scheduling problems under learning and deterioration effects and they showed that single machine problems are polynomially solvable if the performance criterion is makespan, total completion time, total weighted completion time or maximum lateness. Toksarı (2011) presented a single machine problem with the unequal release times under effects of learning and deteriorating simultaneously when the objective is to minimise the makespan. Toksarı (2011) introduced a branch-and-bound algorithm for lower bounds of optimal solutions and proposed a heuristic method. For the readers, some recent papers were studied by Fan et al. (2017), Wang et al. (2017), Azadeh et al. (2017), Mohammadi and Khalilpourazari (2017), Lu (2016), Soroush (2016), Yusriski, Sukoyo, and Halim (2016) and Cheng et al. (2011).

Scheduling problems under learning and/or deterioration effects have been interested for a long time with a considerable number of papers in the literature. There are few researches interested in scheduling problems under learning and/or deterioration effects with fuzzy parameters. The most prominent papers belong to Ahmadizar and Hosseini (2011, 2013), Yeh et al. (2014), Mazdeh, Zaerpour, and Jahantigh (2010), Bentrcia and Mouss (2016) and Rostami, Pilerood, and Mazdeh (2015).

In parallel machine scheduling problems, some significant researches make a good lead for researchers interested in learning and/or deterioration effects. Toksarı and Güner (2008) introduced a MINLP for parallel machine scheduling problems under learning and deterioration effects simultaneously with sequence-dependent setup times and a common due date for all jobs Toksarı and Güner (2009) investigated a parallel machine scheduling problem under position dependent learning effect and non-linear deterioration effect when the objective is to minimise total earliness/tardiness cost. They proved the optimal schedule of their problem is V-shaped. In another study of Toksarı and Güner (2010a), they investigated a parallel machine scheduling problem when the objective is to minimise total earliness/tardiness under effect of time dependent learning and linear/non-linear deterioration with common due date for all jobs. They also showed the optimal schedule is V-shaped for their earliness/tardiness problem. Furthermore, Toksarı and Güner (2010b) proved that the optimal schedule is V-shaped for parallel machine earliness/tardiness problem under position dependent learning and linear deterioration effect with sequence dependent setup times and a common due date.

Although, job deterioration and learning effect are usually defined in deterministic form in the literature, scheduling environment may chance the impact degree of both effects, i.e. a system or a worker may not learn constantly or deterioration level may chance because of some work place conditions such as noise, vibration, temperature, pressure and air flow rate. Does a worker/ system continue to learn constantly or not? How much is the difference between current learning amount and next learning amount going to be? Will deterioration level be changed due to work place conditions? These questions in DM's mind cause vagueness and imprecision for job deterioration and learning effects. Therefore, job deterioration and learning effect may be defined on a closed interval using fuzzy numbers. Fuzzy processing times have been already considered as a part of scheduling environments by researchers (Chanas and Kasperski 2003; Niu, Jiao, and Gu 2008; Sakawa and Kubota 2000; Sakawa and Mori 1999; Yeh et al. 2014). The most common real life example (see Gupta and Gupta 1988) for deterioration effect is steel rolling process for ingots. The ingots have to be processed on rolling machine with a certain temperature. If temperature of that ingot is decreased while waiting in the queue because of any external factors such as air flow rate and weather conditions in the work place, workers take this ingot from the line and reheat it again. Each repeat of taking ingots from the line may lead a learning effect. How much external effects will decrease the temperature of an ingot is unknown and there is an uncertainty for deterioration effect for each job. Uncertain deterioration effect due to unknown work-piece temperature causes uncertain learning amount because each deteriorated ingot's or cooled ingot's temperature is different. In this condition, workers face similar job tasks with different temperatures. This leads uncertainty in learning effect. Both of uncertainty in learning and deterioration effect can be encoded with fuzzy sets in order to illustrate satisfaction levels of DM or possibility of an event such as deteriorating a task or learning. In this example, processing time can be considered as fuzzy numbers because of up/down in machine speed and breakdowns. Thus, a scheduling problem can be considered in a fully fuzzy environment. As a novel, this paper considers all parameters in a scheduling environment as uncertain. The uncertainty in parameters does not depend on the frequency of an event, the uncertainty occurs because the parameters cannot be measured well. While dealing with such an uncertainty that does not depend on frequency of an event, fuzzy sets can be used in order to encode uncertainty. Furthermore, this paper presents a heuristic method (the serpentine algorithm) for parallel scheduling problems when the objective is to minimise the makespan under effects of learning and deterioration.

2. Basic definition of fuzzy set theory

In this section, before building our mathematical model and introducing solution approaches, we introduce some necessary basic definitions of Fuzzy set theory.

In this paper, all parameters are designed in the form of triangular fuzzy numbers. A triangular fuzzy number $\tilde{A}$ with a membership function $\mu_{\tilde{A}}(x) : R \to [0,1]$ is presented with three point on the real axis such that $\tilde{A} = \left(A^L, A^C, A^R\right)$. Membership function of triangular fuzzy number $\tilde{A}$ can be expressed by Equation (1).

$$\mu_{\tilde{A}}^{(x)} = \begin{cases} L_A(x) = \frac{x-A^L}{A^C-A^L}, & \text{if } A^C > x \geq A^L, \\ 1, & \text{if } A^C = x, \\ R_A(x) = \frac{A^R-x}{A^R-A^C}, & \text{if } A^R \geq x > A^C, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Expected value of a triangular fuzzy number $\tilde{A}$ can be shown in Equation (2).

$$E(\tilde{A}) = \frac{A^L + 2A^C + A^R}{4}. \tag{2}$$

3. Fuzzy multi-objective non-linear programming model

Before bulding our proposed mathematical model, learning and deterioration effects must be defined. Biskup (1999) introduced position-based learning effect. Let P_r be basic processing time of the job scheduled at position r, then the actual processing time $P_{[r]}$ of the job at position r under position-dependent learning effect is defined as follows:

$$P_{[r]} = P_r r^a, \tag{3}$$

where $-1 \leq a \leq 0$ is learning effect. Another kind of learning effect is time-dependent learning effect and it was introduced by Kuo and Yang (2006) as follows:

$$P_{[r]} = P_r \left(1 + \sum_{k=1}^{r-1} P_{[k]}\right)^a, \tag{4}$$

where $(1 \leq k \leq r-1)P_{[k]}$ denotes each earlier job's actual processing time before position r. Gupta and Gupta (1988) and Browne and Yechiali (1990) independently studied deterioration effect in scheduling problems. Alidaee and Womer (1999) classified job deterioration effects into three types. These are linear, pairwise linear and non-linear, respectively. Mosheiov (1991) investigated linear job deterioration effect and showed that actual processing time of each job grows linearly with its starting time. Mosheiov (1991) denoted actual processing time under linear job deterioration effect as follows:

$$P_{[r]} = P_r + B * C_{r-1}, \tag{5}$$

where $B > 0$ is linear job deterioration effect introduced by Mosheiov (1991) and C_{r-1} is actual completion time of the job at position $r - 1$. For a job at position r, actual processing time under non-linear job deterioration is defined as follows:

$$P_{[r]} = P_r + B * C_{r-1}^{\beta}, \tag{6}$$

where $\beta > 0$ is non-linear job deterioration effect (see Alidaee and Womer 1999). In this study, we consider four different types of models for calculating actual processing time under different types of learning and deterioration effects. These are;

Model 1: under linear job deterioration and position-dependent learning effects.
Model 2: under linear job deterioration and time-dependent learning effects.
Model 3: under non-linear job deterioration and position-dependent learning effects.
Model 4: under non-linear job deterioration and time-dependent learning effects.

Four actual processing time calculations for all models can be seen in Equations (7)–(10) respectively.

$$P_{[r]} = (P_r + B * C_{r-1})r^a, \tag{7}$$

$$P_{[r]} = (P_r + B * C_{r-1}^{\beta})r^a, \tag{8}$$

$$P_{[r]} = (P_r + B * C_{r-1})\left(1 + \sum_{k=1}^{r-1} P_{[k]}\right)^a, \tag{9}$$

$$P_{[r]} = \left(P_r + B * C_{r-1}^{\beta}\right)\left(1 + \sum_{k=1}^{r-1} P_{[k]}\right)^a. \tag{10}$$

Using Equations (7)–(10), we can build our fuzzy multi-objective mathematical model as shown below. We consider a due date assignment problem on parallel machines with fuzzy processing times under effects of fuzzy learning and deterioration. There are n jobs waiting to be processed on m identical parallel machines. We assumed that all jobs are released at the same time. Preemption and arbitrary idle times are not allowed. Decision variables such as completion times, earliness, tardiness and due dates are in form of triangular fuzzy numbers because processing times of jobs, learning and deterioration effect coefficients are in form of triangular fuzzy numbers.

Indexes

i	index for jobs, $i = 1, \ldots, n$
j	index for machines, $j = 1, \ldots, m$
r	index for positions, $r = 1, \ldots, n$

Parameters

$\tilde{P}_i$	basic fuzzy processing time of job i
$\tilde{a}$	fuzzy learning effect coefficient
$\tilde{B}$	fuzzy linear deterioration effect coefficient
$\tilde{\beta}$	fuzzy non-linear deterioration effect coefficient
MA	earliness penalty cost coefficient for all jobs
MB	tardiness penalty cost coefficient for all jobs
MC	cost of setting due date for all jobs

Decision variables

$\tilde{P}_{[r],j}$	actual processing time of the job at position r on machine j
$\tilde{C}_{[r],j}$	actual completion time of the job at position r on machine j

$\tilde{E}_{[r],j}$ earliness of the job at position r on machine j
$\tilde{T}_{[r],j}$ tardiness of the job at position r on machine j
$\tilde{D}_{[r],j}$ due date of the job at position r on machine j
$\tilde{C}_i$ actual completion time of job i
$\tilde{E}_i$ earliness of job i
$\tilde{T}_i$ tardiness of job i
$\tilde{D}_i$ due date of job i
$x_{i,r,j} \in \{0,1\}$ if job i is assigned at position r on machine j, $x_{i,r,j} = 1$, otherwise it is zero

Objective function

$$\min z = MA * \sum_{i=1}^{n} \tilde{E}_i + MB * \sum_{i=1}^{n} \tilde{T}_i + MC * \sum_{i=1}^{n} \tilde{D}_i. \tag{11}$$

Constraints

$$\tilde{P}_{[r],j} = \sum_{i=1}^{N} \left[x_{i,r,j} * \left[\left(\tilde{P}_i\right) + \left(\tilde{B}\right) * \tilde{C}_{[r-1],j} \right] * r^{\tilde{a}} \right] \quad \forall r,j, \tag{12a}$$

$$\tilde{P}_{[r],j} = \sum_{i=1}^{N} \left[x_{i,r,j} * \left[\left(\tilde{P}_i\right) + \left(\tilde{B}\right) * \tilde{C}_{[r-1],j}^{\tilde{\beta}} \right] * r^{\tilde{a}} \right] \quad \forall r,j, \tag{12b}$$

$$\tilde{P}_{[r],j} = \sum_{i=1}^{N} \left[x_{i,r,j} * \left[\left(\tilde{P}_i\right) + \left(\tilde{B}\right) * \tilde{C}_{[r-1],j} \right] * \left(1 + \sum_{k=1}^{r-1} \tilde{P}_{[k],j} \right)^{\tilde{a}} \right] \quad \forall r,j, \tag{12c}$$

$$\tilde{P}_{[r],j} = \sum_{i=1}^{N} \left[x_{i,r,j} * \left[\left(\tilde{P}_i\right) + \left(\tilde{B}\right) * \tilde{C}_{[r-1],j}^{\tilde{\beta}} \right] * \left(1 + \sum_{k=1}^{r-1} \tilde{P}_{[k],j} \right)^{\tilde{a}} \right] \quad \forall r,j, \tag{12d}$$

$$\tilde{C}_{[0],j} = 0 \quad \forall j, \tag{13}$$

$$\tilde{C}_{[r],j} = \tilde{C}_{[r-1],j} + \tilde{P}_{[r],j} \quad \forall r,j, \tag{14}$$

$$\tilde{C}_{[r],j} + \tilde{E}_{[r],j} - \tilde{T}_{[r],j} = \tilde{D}_{[r],j} \quad \forall r,j, \tag{15}$$

$$\tilde{C}_i + \tilde{E}_i - \tilde{T}_i = \tilde{D}_i \quad \forall i, \tag{16}$$

$$\tilde{C}_i = \sum_{r}^{n} \sum_{j}^{m} x_{i,r,j} * \tilde{C}_{[r],j} \quad \forall i, \tag{17}$$

$$\tilde{D}_i = \sum_r^n \sum_j^m x_{i,r,j} * \tilde{D}_{[r],j} \quad \forall i, \tag{18}$$

$$\tilde{E}_i = \sum_r^n \sum_j^m x_{i,r,j} * \tilde{E}_{[r],j} \quad \forall i, \tag{19}$$

$$\tilde{T}_i = \sum_r^n \sum_j^m x_{i,r,j} * \tilde{T}_{[r],j} \quad \forall i, \tag{20}$$

$$\sum_i^n x_{i,r,j} \leq 1 \quad \forall r, j, \tag{21}$$

$$\sum_r^n \sum_j^m x_{i,r,j} = 1 \quad \forall i, \tag{22}$$

$$x_{i,r,j} + \sum_{I=1}^n x_{I,r+1,j} \leq 2 \quad (I \neq i), \quad (r = 1, 2, \ldots, n-1) \quad \forall i, j, \tag{23}$$

$$x_{i,r+1,j} \leq x_{I,r,j} (I \neq i) \quad (r = 1, 2, \ldots, n-1) \quad \forall i, j, \tag{24}$$

$$\tilde{P}_{[r],j}, \tilde{C}_{[r],j}, \tilde{E}_{[r],j}, \tilde{T}_{[r],j} \geq 0 \quad \forall r, j, \tag{25}$$

$$\tilde{C}_i, \tilde{E}_i, \tilde{T}_i, \tilde{D}_i \geq 0 \quad \forall i, \tag{26}$$

$$x_{i,r,j} \in \{0, 1\} \quad \forall i, r, j. \tag{27}$$

The objective function in Equation (11) is to minimise a cost combination of earliness, tardiness and setting due dates. We introduced four different equations in Equations (7)–(10) for all combinations of learning and deterioration effects. Equations (12a)–(12d) show the calculation of actual processing time of the job assigned at position r on machine j under any types of learning and deterioration effects. Equation (12) shows that all machines are ready at the beginning and scheduling start time is equal and zero for all machines. Equation (14) shows that current job actual completion time is sum of previous job completion time and current job actual processing time. Equations (15) and (16) show that sum of completion time and earliness of a job must be equal to sum of due date and tardiness of that job. Equations (17)–(20) are decision variable transformations between job index and both position and machine indexes. Equation (21) ensures that only one job can be assigned at any position on any machine. Equation (22) guaranties that each job must be assigned at a single position of a machine. Equation (23) shows that each job must be processed for once. Equation (24) ensures that only one other job may come instantly after each job's completion. Equations (25) and (26) show all fuzzy decision variables are non-negative. Equation (27) $x_{i,r,j}$ is a binary decision variable.

4. Solution approaches

In this section, we propose a local search algorithm in order to generate an approximate and faster solution for the problem in this study. To illustrate the performance of proposed local search method, we solve our problem with different solution techniques in the literature. These are fully fuzzy linear programming (FFLP), possibilistic linear programming

(PLP) and fuzzy chance constrained programming techniques. As seen in Section 3, fuzzy constraints are equality constraints. Therefore, we select the adaptive solution techniques for fuzzy equality constraints. Among FFLP solution methods, we select methods of Allahviranloo, Lotfi, and Kiasary (2008), Kumar, Kaur, and Singh (2010, 2011) and Jayalakshmi and Pandian (2012). As PLP methods, we use methods of Lai and Hwang (1992) and Fullér (1986). Furthermore, credibility-based fuzzy chance constraint technique (1998) is used to compare with our proposed method.

4.1 *Proposed local search algorithm*

Defuzzification methods and/or using expected values of fuzzy numbers are common ways to cope with fuzziness. There are five different solution approaches introduced above for solving fuzzy mathematical programming. All of these approaches need lot of CPU time when the problem size raise. Therefore, we propose a local search algorithm dependent on expected values of fuzzy parameters in order to obtain an approximate solution for the problem in this study. Firstly, each fuzzy parameter is converted to its expected value using Equation (2).

Local search is a metaheuristic method whose outputs are obtained fast and efficient for optimisation problems. Local search method searches candidate solutions iteratively by comparing the best solution, obtained before, with each iteration's solution in order to improve its best solution at each iteration. In order to obtain an initial solution for running proposed local search method, we use an algorithm that we called 'The Serpentine Algorithm' based on SPT and equal work position number for all machines as follows:

Step 1: Calculate r, position number, to load equal number of jobs on machines as follows:

n: total job number
m: total machine number
$r = \lceil \frac{n}{m} \rceil$.

Step 2: Sort $E(\tilde{P}_i)$ values in ascending order by introducing new $A(k)$ such as:

$$A(k) = \{\min(E(\tilde{P}_i)), \ldots, \max(E(\tilde{P}_i))\} \quad i = 1, 2, 3, \ldots, n \text{ and } k = 1, 2, 3, \ldots, n$$

Step 3: Assign $A(1)$ at first position of first machine, then continue to assign jobs $A(k)$ until last machine with highest index. After completing assigned jobs at position $r = 1$ for all machines, do same assignment in ascending order of processing times for $r = 2$ from last machine index to first machine index. This process goes on until last job assignment. Figure 1 illustrates this step for 10 jobs and 3 machines and there is the pseudo code for these steps in Figure 2.

After obtaining an initial solution by the serpentine algorithm, we need to improve this solution by assuming this problem objective is to minimise the maximum completion time. If we decrease total completion time of schedule and try to load equal work amount for each machine, this causes a tighter schedule. In order to decrease total completion time and load equal work amount for each machine, there are five swap rules introduced in Figure 3, respectively. If a swap operation doesn't cause an increase in total completion time of the schedule, then that swap operation is applied.

In order to process swap operations introduced in Figure 3 between alternative solutions, we need an algorithm that searches every possible matches for swapping. A pseudo code for swap operations' algorithm is illustrated in Figure 4.

The algorithm in Figure 4 uses expected values of fuzzy processing times without effects of fuzzy deterioration and learning in order to minimise the makespan of the schedule. As a result of above algorithm, we have each ith job's

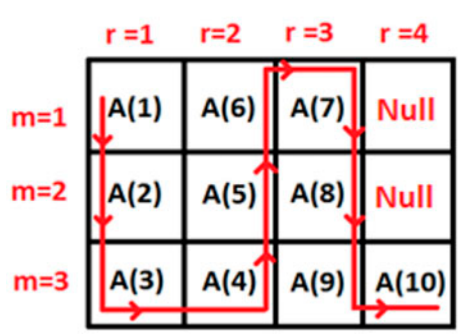

Figure 1. Assignment by the serpentine algorithm.

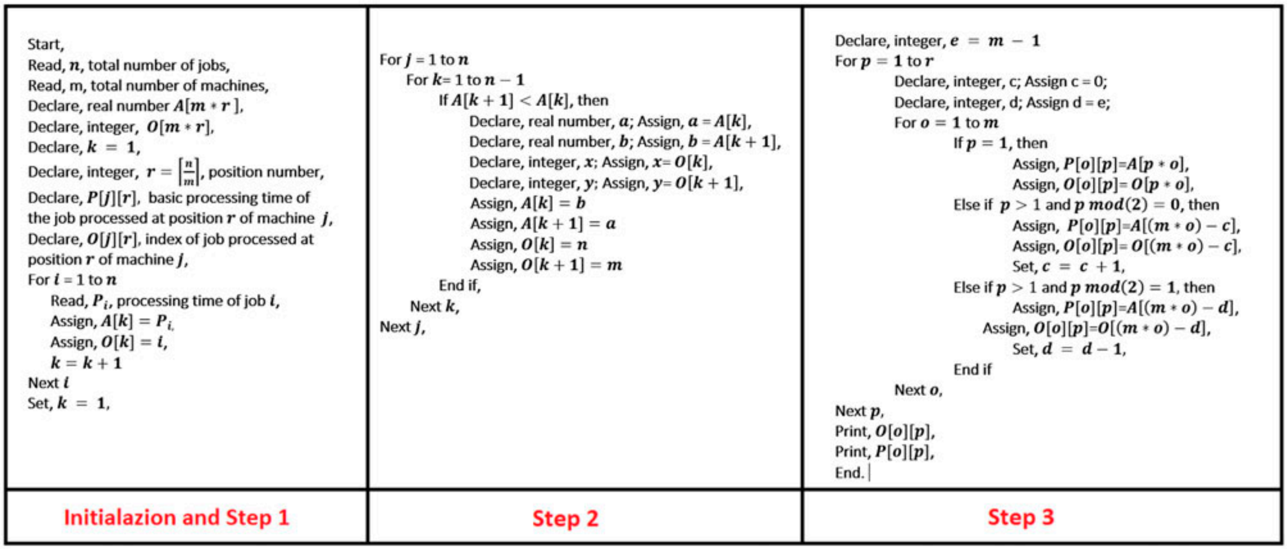

Figure 2. Pseudo code for the serpentine algorithm.

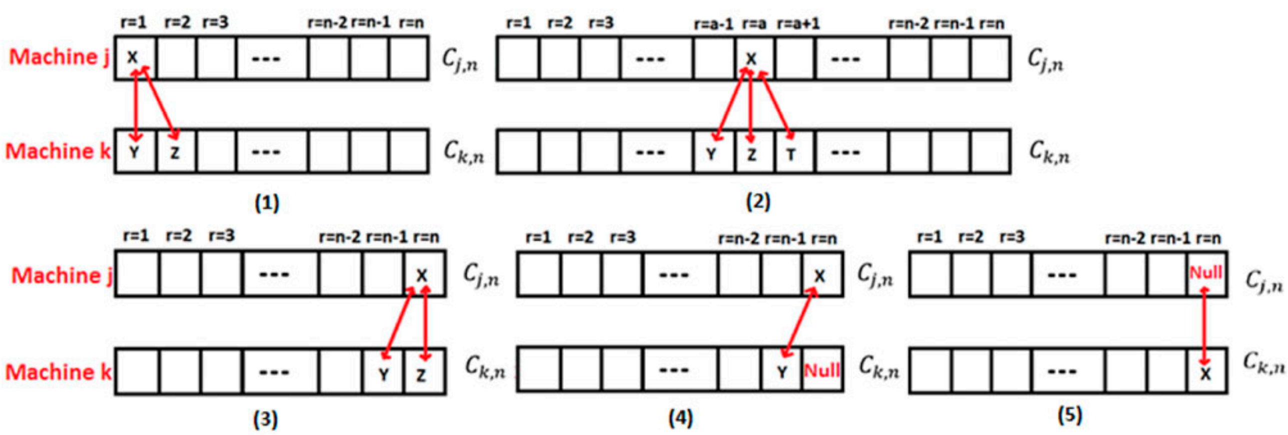

Figure 3. Illustrations of swap rules in local search algorithm.

location on the schedule and its basic processing time. In order to assign job's due dates and determine the objective value, we need another algorithm that assuming $D_i = C_i$ suggested by Baker and Trietsch (2009) for setting cost of earliness/tardiness as zero. The assumption of $D_i = C_i$ directly denotes that the objective function of the problem turns into $MC * \sum C_i$. Next phase of finding an approximate solution for the problem is setting of early/tardy jobs and due-dates. After all swap operations, completion times and processing times are recalculated by considering learning and deterioration effects for each model as follows:

$$P^*_{[j],[r]} = \begin{cases} \left[\left(P_{[j][r]}\right) + \left(E\left(\widetilde{B}\right)\right) * C^*_{[j][r-1]}\right] * r^{E\left(\widetilde{a_r}\right)} \quad \forall r,j & \text{if model 1 is selected,} \\ \left[\left(P_{[j][r]}\right) + \left(E\left(\widetilde{B}\right)\right) * {C^*_{[j][r-1]}}^{E\left(\widetilde{\beta}\right)}\right] * r^{E\left(\widetilde{a_r}\right)} \quad \forall r,j & \text{if model 2 is selected,} \\ \left[\left(P_{[j][r]}\right) + \left(E\left(\widetilde{B}\right)\right) * C^*_{[j][r-1]}\right] * \left(1 + \sum_{k=1}^{r-1} P_{[j],[k]}\right)^{E\left(\widetilde{a_r}\right)} \quad \forall r,j & \text{if model 3 is selected,} \\ \left[\left(P_{[j][r]}\right) + \left(E\left(\widetilde{B}\right)\right) * {C^*_{[j][r-1]}}^{E\left(\widetilde{\beta}\right)}\right] * \left(1 + \sum_{k=1}^{r-1} P_{[j],[k]}\right)^{E\left(\widetilde{a_r}\right)} \quad \forall r,j & \text{if model 4 is selected,} \end{cases} \tag{28}$$

```
Start,
Read, n, number of jobs,
Read, m, number of machines,
Read, r, number of positions,
Read, O[m][r], job index of the job at position r of machine m,
Read, P[m][r], basic processing time of the job at position r of machine m,
Declare, integer, k = r, number of iteration,
For p = 1 to k
```

Initialization

```
For x = 1 to m
    For y = 1 to m
        If not x= y, then
            Calculate C_max,
            Calculate C_r,N, / The completion time
            of last job of each machine /,
            Apply Swap Rule 1,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 2,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 3,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 4,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 5,
        End if
    Next y
Next x
```

1st Swap Direction

```
For x = m to 1 step x = x − 1
    For y = 1 to m
        If not x= y, then
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 1,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 2,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 3,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 4,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 5,
        End if
    Next y
Next x
```

2nd Swap Direction

```
For x = m to 1 step x = x − 1
    For y = m to 1 step y = y − 1
        If not x= y, then
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 1,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 2,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 3,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 4,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 5,
        End if
    Next y
Next x
```

3rd Swap Direction

```
For x = 1 to m
    For y = m to 1 step y = y − 1
        If not x= y, then
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 1,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 2,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 3,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 4,
            Calculate C_max, Calculate C_r,N for all machines,
            Apply Swap Rule 5,
        End if
    Next y
Next x
```

4th Swap Direction

```
Next p
Print, O[m][r],
Print, P[m][r],
End.
```

End of Algorithm

Figure 4. Pseudo code for swap operations' algorithm.

$$C^*_{[j][0]} = 0 \quad \forall j, \tag{29}$$

$$C^*_{[j][r]} = P^*_{[j][r]} + C^*_{[j][r-1]} \quad \forall r, j, \tag{30}$$

where $C^*_{[j][r]}$ is expected actual completion time and $P^*_{[j][r]}$ is expected actual processing time under effects of learning and deterioration. Due to all penalty costs are same for all jobs, any policy for any job can apply for all schedule i.e. all jobs are tardy or early or completed on their due dates. Setting all due dates as $D_i = C_i$ is an initial solution for the problem. The objective functions in Equation (31) represent three different situations in which all jobs completed on time, tardy and early, respectively.

$$f = \begin{cases} f_1 = MC * \sum D_i & \text{if } D_i = C_i, \\ f_2 = MC * \sum \underline{D}_i + MB * \sum (C_i - \underline{D}_i) & \text{if } D_i < C_i \quad \text{and} \quad D_i = \underline{D}_i, \\ f_3 = MC * \sum \bar{D}_i + MA * \sum (\bar{D}_i - C_i) & \text{if } D_i > C_i \quad \text{and} \quad D_i = \bar{D}_i. \end{cases} \tag{31}$$

In Equation (31), objective function f_1 denotes that all jobs' due dates equal to their completion times obtained from Equations (28)–(30). f_2 is an objective function when all jobs completed early and f_3 is for the case in which all jobs are tardy. Figure 5 illustrates all situations and it shows that $\underline{D}_i$ value tends to infinity. On the contrary, $\bar{D}_i$ can just get values between zero and C_i.

Theorem 1: Although early penalty cost is the lowest, setting all jobs early does not improve the objective function of the schedule obtained after swap operations. By ignoring early penalty and selecting less penalty type, *DM* should set all jobs' situation as tardy or on time. There are two different cases; these are $MA < MB < MC$ and $< MC < MB$, respectively. In case of $MA < MB = MC$, *DM* may choose *MB* or *MC*.

Proof 1: In case of $MA < MB < MC$, let A is the objective function value except assigning the last job on any machine. S is the schedule that the last job is tardy and S' is the schedule that the last job is early. If objective function $f(S)$ is less then $f(S')$, then S schedule dominates S' schedule.

$$f(S) = [A + MB * (C_i - \underline{D}_i) + MC * \underline{D}_i],$$

$$f(S') = [A + MA * (\bar{D}_i - C_i) + MC * \bar{D}_i],$$

$$f(S') - f(S) = [A + MA * (\bar{D}_i - C_i) + MC * \bar{D}_i] - [A + MB * (C_i - \underline{D}_i) + MC * \underline{D}_i],$$

$$f(S') - f(S) = MA * \bar{D}_i - MA * C_i + MC * \underline{D}_i - MB * C_i + MB * \underline{D}_i. \tag{32}$$

Equation (32) may be rewritten using $\bar{D}_i = e + C_i$ and $\underline{D}_i = C_i - t$ as follows:

$$f(S') - f(S) = e * (MA + MC) + C_i * (MA + MC) - C_i * (MA + MB) + t * (MC - MB) - C_i * (MC - MB),$$

$$f(S') - f(S) = e * (MA + MC) + t * (MC - MB),$$

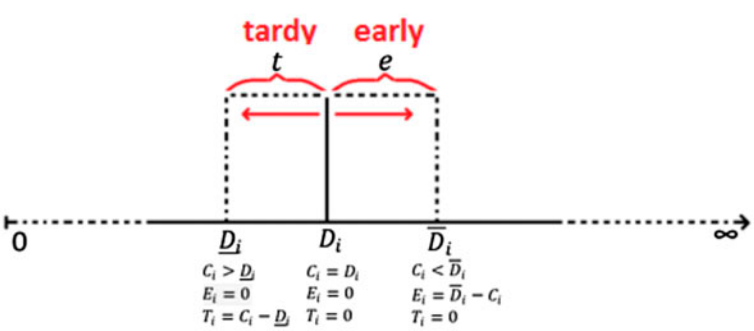

Figure 5. Tardy, early and on time completion of jobs.

where $e > 0$ and $t > 0$ are any earliness and tardiness values for the last job. Since $MC > MB$, $MA + MC > 0$, $e > 0$ and $t > 0$. In case of $MA < MB < MC, f(S') - f(S)$ is greater than zero and this denotes that S schedule dominates S'.

In case of $MA < MC < MB$, let S is the schedule that the last job is completed on its due date and S' is the schedule that the last job is early. If objective function $f(S)$ is less then $f(S')$, then S schedule dominates S' schedule.

$$f(S') = [A + MA * (\bar{D}_i - C_i) + MC * \bar{D}_i],$$

$$f(S) = [A + MC * D_i],$$

$$f(S') - f(S) = [A + MA * (\bar{D}_i - C_i) + MC * \bar{D}_i] - [A + MC * D_i]. \tag{33}$$

Equation (33) is rewritten using $D_i = C_i$ as follows:

$$f(S') - f(S) = MA * \bar{D}_i - MA * C_i + MC * \bar{D}_i - MC * C_i,$$

$$f(S') - f(S) = \bar{D}_i * (MA + MC) - C_i * (MA + MC),$$

$$f(S') - f(S) = (MA + MC) * (\bar{D}_i - C_i). \tag{34}$$

Since $\bar{D}_i > C_i$ and $MA + MC > 0$, Equation (34) illustrates that $f(S') - f(S) > 0$ and S schedule dominates S' schedule in case of $MA < MC < MB$. Thus, Theorem 1 is proved for each situation.

Theorem 2: Although, early penalty cost is the smallest positive value among objective function coefficients, setting all jobs' situation as early does not improve the objective function value. This situation is defined and proved in Theorem 1 and Proof 1. By ignoring early penalty cost, selecting small cost coefficient and setting all jobs' situation as that cost type is more efficient solution for our problem, i.e. if tardy penalty is less than completing job on time, then setting all jobs' due dates as zero and tardiness for all job as $T_i = C_i \forall i$ gives more efficient solution for our problem. On the contrary, if MC < MB, then setting all jobs' due dates as $D_i = C_i \forall i$ gives more efficient solution.

Proof 2: In case of $MB < MC$, let S is the schedule that last job is tardy and S' is the schedule that the last job is completed on time. If the objective function of S is smaller than S''s objective function, then S dominates S'.

$$f(S') = [A + MC * D_i],$$

$$f(S) = [A + MC * \underline{D}_i + MB * (C_i - \underline{D}_i)],$$

$$f(S') - f(S) = [A + MC * D_i] - [A + MC * \underline{D}_i + MB * (C_i - \underline{D}_i)]. \tag{35}$$

If we regulate Equation (35) using $\underline{D}_i = C_i - t$ and $D_i = C_i$ as follows:

$$f(S') - f(S) = MC * C_i - MC * (C_i - t) - MB * (C_i - C_i + t),$$

$$f(S') - f(S) = (MC - MB) * t.$$

Since $MB < MC$, $f(S') - f(S)$is greater than zero and it proves that S schedule dominates S' schedule. In case of $MB > MC$, S' schedule dominates S schedule and setting all jobs' due dates as $D_i = C_i$ gives more efficient solution.

Using Theorems 1 and 2, the due date assignment algorithm has four steps as follows:

Step 1: Read all necessary values (n, m, r) and $P[j][r]$ outputs from swap operations' algorithm.

Step 2: Using Equations (28)–(30), calculate $P^*[j][r]$ and $C^*[j][r]$. Create the initial solution providing that $D_i = C_i \forall i$.

Step 3: Using Theorems 1 and 2, assign jobs' due dates.

Step 4: Using Equation (31), calculate the objective value of the problem.

5. Numerical examples

In this section, we illustrate some numerical examples with different sets of objective coefficients. For *MA*, *MB* and *MC* values, respectively, these sets are (10, 5, 3), (10, 3, 5), (5, 10, 3), (5, 3, 10), (3, 10, 5), (3, 5, 10) and (3, 3, 3). Fuzzy processing times for 10 jobs are shown in Table 1. Firstly, we examine all of solution approaches introduced in Section 4, with 10 jobs and 2 parallel machines. Mathematical models were solved by DICOPT solver in Gams 21.6 software using a standard desktop computer that has CPU of Intel Core i5, 2.8 GHz and 4 GB RAM. The proposed local search algorithm was coded and solved using same computer with Excel 2010 VBA. For all numerical examples, fuzzy learning effect, linear deterioration effect and non-linear deterioration effect are $\tilde{a} = \{-0.018, -0.016, -0.014\}$, $\tilde{B} = \{0.06, 0.08, 0.09\}$ and $\tilde{\beta} = \{0.1, 0.2, 0.3\}$, respectively.

There are seven different sets of objective coefficients and four different models. In order to compare solution approaches, we present results according to all models in Table 2.

As shown in Table 2, all results of solution approaches are close to each other for each model. Since methods of deffuzzification and copying with fuzziness used in solution approaches are different than each other and the problem is a MINLP with non-linear functions, we cannot expect that solution approaches' results are the same and all methods give the same schedule. The ranking method of Allahviranloo, Lotfi, and Kiasary's (2008) approach is similar to the expected value of fuzzy numbers used in our proposed local search method. The method of Kumar, Kaur, and Singh's (2010, 2011) produces better objective function values for some problems. The main reason of these differences between Kumar, Kaur, and Singh's (2010, 2011) method and others including local search is that Kumar, Kaur, and Singh's (2010, 2011) method considers only expected value of a fuzzy objective function and all fuzzy constraints in their method are executed for three times for left, peak and right values of fuzzy parameters in a single model. Kumar, Kaur, and Singh's (2010, 2011) method searches the best solution by considering only deffuzzied decision variables in objective function. Our proposed local search method deffuzzies all fuzzy parameters at the beginning. For a triangular fuzzy number such as {10, 22, 32}, local search and Allahviranloo, Lotfi, and Kiasary's (2008) method produce a number of 21.5. Fullér (1986) and Liu and Iwamura's (for $\alpha = 0.5$) methods produce 23.33 and 22 for the same fuzzy number, respectively. In Lai and Hwang's (1992) method, decison variables are deterministic and rest of all parameters in a standard linear programming model are in form of triangular fuzzy numbers with triangular possibility distributions. We expanded Lai and Hwang's (1992) method by considering all parameters and decision variables in form of triangular fuzzy numbers. Lai and Hwang's (1992) (for $\beta = 0.5$) method produces another triple illustration of the same triangular fuzzy number as {16, 22, 27} and solve the original problem at least seven times in order to maximise level of interaction among negative and positive ideal solutions of sub three problems of the original problem. On the other hand, Kumar, Kaur, and Singh (2010, 2011) uses the same deffuzzfication method for only objective function of the problem. In Jayalakshmi and Pandian's (2012) method executes the original model three times for left, peak and right values of fuzzy parameters. Furthermore, there is no deffuzzfication or crisp equality for fuzzy constraints and numbers in their method. The differences in objective function values can be explained with the differences in solution approaches. The objective function value of the best schedule obtained proposed local search can be less or more than other method's objective function values because of these differences, if we calculate the same schedule using other methods. Therefore, we can say that the performance of proposed local search method for each model seems good enough to cope with large scale problems by comparing our method and Allahviranloo, Lotfi, and Kiasary's (2008) method because both of two method uses the same deffuzzication method. The results in Table 2 can be used for future comparisons of the readers who want to repeat this study or want to put forward a better solution algorithm. Table 3 illustrates CPU times in

Table 1. Fuzzy processing times for numerical examples with 10 jobs/2 machines.

i	$\tilde{P}_i = \{P_i^L, P_i^C, P_i^R\}$
1	{9, 13, 14}
2	{13, 17, 19}
3	{15, 16, 19}
4	{8, 13, 16}
5	{10, 13, 16}
6	{9, 12, 14}
7	{11, 13, 16}
8	{12, 14, 18}
9	{20, 23, 26}
10	{11, 13, 16}

Table 2. Results of solution approaches according to each model.

10 Jobs/2 machines	Model type	$MA=10$, $MB=5$, $MC=3$	$MA=10$, $MB=3$, $MC=5$	$MA=5$, $MB=10$, $MC=3$	$MA=5$, $MB=3$, $MC=10$	$MA=3$, $MB=10$, $MC=5$	$MA=3$, $MB=5$, $MC=10$	$MA=3$, $MB=3$, $MC=3$
Allahviranloo, Lotfi, and Kiasary(2008)	Model-1	1320.956	1339.084	1312.532	1327.907	2184.048	2184.048	1310.429
Kumar, Kaur, and Singh (2010, 2011)		1318.108	1318.108	1318.108	1318.108	2205.597	2196.847	1318.108
Jayalakshmi and Pandian(2012)		1333.809	1333.809	1333.809	1333.809	2228.773	2223.014	1333.809
Lai and Hwang (1992) ($\gamma = 0.5$)		1391.596	1435.803	1388.836	1416.274	2326.881	2341.515	1388.836
Fullér (1986)		1321.172	1321.172	1326.247	1321.172	2208.651	2201.953	1321.172
Fuzzy chance cons. (Liu and Iwamura 1998) ($\alpha = 0.5$)		1333.809	1333.809	1333.809	1336.982	2223.014	2228.303	1333.809
Local Search		1321.170	1321.170	1321.170	1321.170	2201.950	2201.950	1321.170
Allahviranloo, Lotfi, and Kiasary(2008)	Model-2	1249.479	1198.458	1198.458	1198.458	1997.429	1997.429	1198.458
Kumar, Kaur, and Singh (2010, 2011)		1207.453	1203.722	1203.722	1203.722	2006.210	2006.203	1203.722
Jayalakshmi and Pandian(2012)		1215.316	1215.316	1215.316	1215.316	2025.527	2025.527	1215.316
Lai and Hwang (1992)($\gamma = 0.5$)		1277.462	1258.383	1258.858	1258.445	2098.097	2119.113	1265.474
Fullér (1986)		1211.490	1207.031	1207.031	1207.031	2017.755	2011.719	1207.031
Fuzzy chance cons. (Liu and Iwamura 1998) ($\alpha = 0.5$)		1215.316	1221.243	1215.316	1215.316	2025.527	2035.404	1215.316
Local Search		1208.161	1208.161	1208.161	1208.161	2013.602	2013.602	1208.161
Allahviranloo, Lotfi, and Kiasary(2008)	Model-3	1263.690	1373.934	1261.575	1258.798	2092.719	2092.7191	1255.632
Kumar, Kaur, and Singh (2010, 2011)		1284.166	1284.152	1284.152	1284.152	2262.132	2140.253	1284.152
Jayalakshmi and Pandian(2012)		1302.256	1302.256	1302.256	1302.256	2164.909	2164.9087	1302.256
Lai and Hwang (1992) ($\gamma = 0.5$)		1352.676	1353.500	1356.853	1353.500	2276.232	2255.848	1352.677
Fullér (1986)		1286.710	1286.710	1286.709	1286.710	2144.516	2144.516	1286.710
Fuzzy chance cons. (Liu and Iwamura 1998) ($\alpha = 0.5$)		1298.945	1298.956	1298.945	1298.945	2164.909	2164.909	1298.945
Local Search		1286.500	1286.500	1286.500	1286.500	2144.166	2144.166	1286.500
Allahviranloo, Lotfi, and Kiasary(2008)	Model-4	1152.890	1155.992	1156.146	1045.825	1926.654	1733.171[a]	1039.90[a]
Kumar, Kaur, and Singh (2010, 2011)		1214.568	1176.254	1178.431	1176.254	1967.801	1960.424	1176.268
Jayalakshmi and Pandian(2012)		1190.155	1190.155	1190.155	1190.155	1978,838	1978.838	1190.155
Lai and Hwang (1992) ($\gamma = 0.5$)		1301.799	1267.576	1260.741	1266.726	2101.370	2117.327	1260.769
Fullér (1986)		1179.136	1179.146	1182.828	1179.136	1965.227	1965.251	1179.136
Fuzzy chance cons. (Liu and Iwamura 1998) ($\alpha = 0.5$)		1187.288	1187.288	1187.303	1187.297	1978.813	1983.542	1187.303
Local search		1176.218	1176.218	1176.218	1176.218	1960.363	1960.363	1176.218

[a]Relaxed solutions.

Table 3. CPU times in seconds of proposed local search method for each model.

# of jobs	# of machines	Time for Model 1 (in s)	Time for Model 2 (in s)	Time for Model 3 (in s)	Time for Model 4 (in s)
10	2	0.06	0.06	0.06	0.08
20	3	0.19	0.19	0.16	0.16
50	4	0.89	0.88	0.89	0.91
100	8	6.86	6.84	6.82	6.86
250	10	51.20	51.88	51.18	51.31
500	20	420.61	437.11	434.26	438.20
750	30	1504.00	1489.77	1519.24	1469.59
1000	40	3342.44	3054.71	2977.96	3125.54

```
Start,
Read, n, number of jobs,
For i = 1 to n
    Assign, P_i^L = Round (100* Random Number in (0,1) ) + 1 ,
    Assign, P_i^C = P_i^L + (Round(10* Random Number in (0,1)) + 1),
    Assign, P_i^R = P_i^C + (Round(10* Random Number in (0,1)) + 1),
Next n
Print, P_i^L, P_i^C, P_i^R,
End.
```

Figure 6. The algorithm for generating fuzzy processing numbers.

seconds for each model with different numbers of jobs and machines. Fuzzy learning and deterioration effects are same fuzzy numbers used previously in this section. For the problems in Table 3; the objective function coefficients *MA*, *MB* and *MC* are 10, 5 and 3, respectively. Fuzzy processing times for these problems are generated randomly except first ten jobs using the algorithm introduced in Figure 6.

For 10 jobs and 2 machines problems, each of fuzzy mathematical programming approaches takes more than forty minutes to find a solution. Proposed local search algorithm takes nearly 0.07 s for finding a solution for the same problem size.

6. Conclusion

In this study, we have studied a multi-objective parallel machine scheduling problem under fully fuzzy environment with fuzzy job deterioration effect, fuzzy learning effect and fuzzy processing times. In our problem, due dates are decision variables and this problem is also due date assignment problem for parallel machines under fuzzy environment. Furthermore, we investigated multiple solution approaches for our problem by comparing their results. Moreover, we proposed a local search method that uses SPT rule of expected values of fuzzy processing time and uses an initial solution algorithm that we called the serpentine algorithm. Numerical examples for different penalties were illustrated in Section 5 and results of different solution approaches are close to each other's. In fact, each solution approach gives an efficient solution for the fuzzy problem but when the problem size raises, execution times of all solution approaches based on mathematical models increase progressively. Therefore, for large scale problems, our proposed local search method may take logical running time by obtaining efficient solutions. Furthermore, the serpentine algorithm and proposed local search can be used for parallel machine scheduling problem when the objective is to minimise the makespan under effects of learning and deterioration. Fuzzification techniques and fuzzy mathematical programming approaches introduced in this paper can be used in order to deal with earliness/tardiness or due date assignment parallel machine scheduling problems when DM's judgement is biased and scheduling parameters cannot be well measured. For future comparisons, results of different solution approaches can be used.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Oğuzhan Ahmet Arık http://orcid.org/0000-0002-7088-2104

References

Abdullah, Salwani, and Majid Abdolrazzagh-Nezhad. 2014. “Fuzzy Job-shop Scheduling Problems: A Review.” *Information Sciences* 278: 380–407. doi:10.1016/j.ins.2014.03.060.

Ahmadizar, Fardin, and Leila Hosseini. 2011. “Single-machine Scheduling with a Position-based Learning Effect and Fuzzy Processing times.” *The International Journal of Advanced Manufacturing Technology* 56 (5–8): 693–698. doi:10.1007/s00170-011-3190-0.

Ahmadizar, Fardin, and Leila Hosseini. 2013. “Minimizing Makespan in a Single-machine Scheduling Problem with a Learning Effect and Fuzzy Processing times.” *The International Journal of Advanced Manufacturing Technology* 65 (1–4): 581–587. doi:10.1007/s00170-012-4198-9.

Alidaee, Bahram, and Norman Keith Womer. 1999. “Scheduling with Time Dependent Processing Times: Review and Extensions.” *Journal of the Operational Research Society* 50 (7): 711–720.

Allahviranloo, T., F. H. Lotfi, and M. K. Kiasary. 2008. “Solving Fully Fuzzy Linear Programming Problem by the Ranking Function.” *Applied Mathematical Sciences* 2 (1): 19–32.

Azadeh, A., H. Habibnejad-Ledari, S. Abdolhossein Zadeh, and M. Hosseinabadi Farahani. 2017. “A Single-machine Scheduling Problem with Learning Effect, Deterioration and Non-monotonic Time-dependent Processing times.” *International Journal of Computer Integrated Manufacturing* 30 (2–3): 292–304. doi:10.1080/0951192X.2016.1145808.

Bachman, A., and A. Janiak. 2004. “Scheduling Jobs with Position-dependent Processing times.” *Journal of the Operational Research Society* 55 (3): 257–264. doi:10.1057/palgrave.jors.2601689.

Bachman, A., A. Janiak, and Mikhail Y. Kovalyov. 2002. “Minimizing the Total Weighted Completion Time of Deteriorating Jobs.” *Information Processing Letters* 81 (2): 81–84. doi:10.1016/S0020-0190(01)00196-X.

Baker, Kenneth R., and Dan Trietsch. 2009. *Principles of Sequencing and Scheduling*. Hoboken, NJ: Wiley.

Behnamian, J. 2016. “Survey on Fuzzy Shop Scheduling.” *Fuzzy Optimization and Decision Making* 15 (3): 331–366. doi:10.1007/s10700-015-9225-5.

Bentrcia, Toufik, and Leila-Hayet Mouss. 2016. “Fuzzy Modeling of Single Machine Scheduling Problems including the Learning Effect.” In *Metaheuristics for Production Systems*, edited by El-Ghazali Talbi, Farouk Yalaoui, and Lionel Amodeo, 315–348. Cham: Springer. doi:10.1007/978-3-319-23350-5_14.

Biskup, Dirk. 1999. “Single-machine Scheduling with Learning Considerations.” *European Journal of Operational Research* 115 (1): 173–178. doi:10.1016/S0377-2217(98)00246-X.

Biskup, Dirk. 2008. “A State-of-the-art Review on Scheduling with Learning Effects.” *European Journal of Operational Research* 188 (2): 315–329. doi:10.1016/j.ejor.2007.05.040.

Browne, Sid, and Uri Yechiali. 1990. “Scheduling Deteriorating Jobs on a Single Processor.” *Operations Research* 38 (3): 495–498.

Chanas, Stefan, and Adam Kasperski. 2003. “On Two Single Machine Scheduling Problems with Fuzzy Processing times and Fuzzy Due Dates.” *European Journal of Operational Research* 147 (2): 281–296. doi:10.1016/S0377-2217(02)00561-1.

Cheng, T. C. E., W.-H. Wu, S.-R. Cheng, and C.-C. Wu. 2011. “Two-agent Scheduling with Position-based Deteriorating Jobs and Learning Effects.” *Applied Mathematics and Computation* 217 (21): 8804–8824. doi:10.1016/j.amc.2011.04.005.

Cheng, T. C. Edwin, Chin Chia Wu, and Wen Chiung Lee. 2008a. “Some Scheduling Problems with Sum-of-processing-times-based and Job-position-based Learning Effects.” *Information Sciences* 178 (11): 2476–2487. doi:10.1016/j.ins.2008.02.002.

Cheng, T. C. Edwin, Chin Chia Wu, and Wen Chiung Lee. 2008b. “Some Scheduling Problems with Deteriorating Jobs and Learning Effects.” *Computers and Industrial Engineering* 54: 972–982. doi:10.1016/j.cie.2009.06.016.

Dubois, Didier, Hélène Fargier, and Henri Prade. 1995. “Fuzzy Constraints in Job-shop Scheduling.” *Journal of Intelligent Manufacturing* 6 (4): 215–234. doi:10.1007/BF00128646.

Eren, T. 2009. “Human and Machine Effects in a Just-in-time Scheduling Problem.” *Human Factors and Ergonomics in Manufacturing* 19 (4): 294–299. doi:10.1002/hfm.20160.

Fan, W., J. Pei, X. Liu, P. M. Pardalos, and M. Kong. 2017. “Serial-batching Group Scheduling with Release times and the Combined Effects of Deterioration and Truncated Job-dependent Learning.” *Journal of Global Optimization*. doi:10.1007/s10898-017-0536-7.

Fullér, Robert. 1986. “On a Special Type of Fuzzy Linear Programming.” *Colloquia Mathematica Societatis Janos Bolyai* 50: 511–519.

Gawiejnowicz, Stanisław. 2007. “Scheduling Deteriorating Jobs Subject to Job or Machine Availability Constraints.” *European Journal of Operational Research* 180 (1): 472–478. doi:10.1016/j.ejor.2006.04.021.

Guo, Peng, Wenming Cheng, and Yi Wang. 2015. "Parallel Machine Scheduling with Step-deteriorating Jobs and Setup times by a Hybrid Discrete Cuckoo Search Algorithm." *Engineering Optimization* 47 (11): 1564–1585. doi:10.1080/0305215X.2014.982634.

Gupta, J. N. D., and S. K. Gupta. 1988. "Single Facility Scheduling with Nonlinear Processing times." *Computers and Industrial Engineering* 14 (4): 387–393.

Han, Sangsu, Hiroaki Ishii, and Susumu Fujii. 1994. "One Machine Scheduling Problem with Fuzzy Duedates." *European Journal of Operational Research* 79 (1): 1–12.

Ishibuchi, Hisao, Naohisa Yamamoto, Tadahiko Murata, and Hideo Tanaka. 1994. "Genetic Algorithms and Neighborhood Search Algorithms for Fuzzy Flowshop Scheduling Problems." *Fuzzy Sets and Systems* 67 (1): 81–100.

Ishii, Hiroaki, and Minoru Tada. 1995. "Single Machine Scheduling Problem with Fuzzy Precedence Relation." *European Journal of Operational Research* 87 (2): 284–288. doi:10.1016/0377-2217(94)00162-6.

Ishii, Hiroaki, Minoru Tada, and Teruo Masuda. 1992. "Two Scheduling Problems with Fuzzy Due-dates." *Fuzzy Sets and Systems* 46 (3): 339–347.

Janiak, A., and R. Rudek. 2009. "Experience-based Approach to Scheduling Problems with the Learning Effect." *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 39 (2): 344–357. doi:10.1109/TSMCA.2008.2010757.

Jayalakshmi, M., and P. Pandian. 2012. "A New Method for Finding an Optimal Fuzzy Solution for Fully Fuzzy Linear Programming Problems." *Ijera.Com* 2 (4): 247–254 http://www.ijera.com/papers/Vol2_issue4/AJ24247254.pdf.

Koulamas, Christos, and George J. Kyparisis. 2007. "Single-machine and Two-machine Flowshop Scheduling with General Learning Functions." *European Journal of Operational Research* 178 (2): 402–407. doi:10.1016/j.ejor.2006.01.030.

Kumar, Amit, Jagdeep Kaur, and Pushpinder Singh. 2010. "Fuzzy Optimal Solution of Fully Fuzzy Linear Programming Problems with Inequality Constraints." *International Journal of Mathematical and Computer Sciences* 6 (1): 37–41. www.waset.org/journals/ijmcs/v6/v6-1-7.pdf.

Kumar, Amit, Jagdeep Kaur, and Pushpinder Singh. 2011. "A New Method for Solving Fully Fuzzy Linear Programming Problems." *Applied Mathematical Modelling* 35 (2): 817–823. doi:10.1016/j.apm.2010.07.037.

Kuo, Wen-Hung, and Dar-Li Yang. 2006. "Minimizing the Total Completion Time in a Single-machine Scheduling Problem with a Time-dependent Learning Effect." *European Journal of Operational Research* 174 (2): 1184–1190. doi:10.1016/j.ejor.2005.03.020.

Kuroda, Mitsuru, and Zeng Wang. 1996. "Fuzzy Job Shop Scheduling." *International Journal of Production Economics* 44 (1–2): 45–51. doi:10.1016/0925-5273(95)00091-7.

Lai, Young-Jou, and Ching-Lai Hwang. 1992. "A New Approach to Some Possibilistic Linear Programming Problems." *Fuzzy Sets and Systems* 49 (2): 121–133.

Lee, Wen Chiung, and Chin Chia Wu. 2004. "Minimizing Total Completion Time in a Two-machine Flowshop with a Learning Effect." *International Journal of Production Economics* 88 (1): 85–93. doi:10.1016/S0925-5273(03)00179-8.

Li, Xuesong, Hiroaki Ishii, and Minghao Chen. 2012. "Batch Scheduling Problem with Due-date and Fuzzy Precedence Relation." *Kybernetika* 48 (2): 346–356.

Li, Xuesong, Hiroaki Ishii, and Minghao Chen. 2015. "Single Machine Parallel-batching Scheduling Problem with Fuzzy Due-date and Fuzzy Precedence Relation." *International Journal of Production Research* 53 (9): 2707–2717.

Li, Xuesong, Hiroaki Ishii, and Teruo Masuda. 2012. "Single Machine Batch Scheduling Problem with Fuzzy Batch Size." *Computers and Industrial Engineering* 62 (3): 688–692.

Liu, Baoding, and Kakuzo Iwamura. 1998. "Chance Constrained Programming with Fuzzy Parameters." *Fuzzy Sets and Systems* 94 (2): 227–237. doi:10.1016/S0165-0114(96)00236-9.

Lu, Y.-Y. 2016. "Research on No-idle Permutation Flowshop Scheduling with Time-dependent Learning Effect and Deteriorating Jobs." *Applied Mathematical Modelling* 40 (4): 3447–3450. doi:10.1016/j.apm.2015.09.081.

Mazdeh, Mohammad Mahavi, Farzad Zaerpour, and Farzad Firouzi Jahantigh. 2010. "A Fuzzy Modeling for Single Machine Scheduling Problem with Deteriorating Jobs." *International Journal of Industrial Engineering Computations* 1 (2): 147–156. doi:10.5267/j.ijiec.2010.02.004.

McCahon, Cynthia S., and E. Stanley Lee. 1990. "Job Sequencing with Fuzzy Processing times." *Computers & Mathematics with Applications* 19 (7): 31–41.

Mohammadi, M., and S. Khalilpourazari. 2017. "Minimizing Makespan in a Single Machine Scheduling Problem with Deteriorating Jobs and Learning Effects." In *ACM International Conference Proceeding Series ICSCA '17 Proceedings of the 6th International Conference on Software and Computer Applications*, Bangkok, Thailand, 310–315. doi:10.1145/3056662.3056715.

Mosheiov, Gur. 1991. "V-shaped Policies for Scheduling Deteriorating Jobs." *Operations Research* 39 (6): 979–991. doi:10.1287/opre.39.6.979.

Mosheiov, Gur. 1995. "Scheduling Jobs with Step-deterioration; Minimizing Makespan on a Single- and Multi-machine." *Computers and Industrial Engineering* 28 (4): 869–879. doi:10.1016/0360-8352(95)00006-M.

Mosheiov, Gur. 1998. "Multi-machine Scheduling with Linear Deterioration." *INFOR: Information Systems and Operational Research* 36 (4): 205–214. doi:10.1080/03155986.1998.11732359.

Mosheiov, Gur. 2001. "Scheduling Problems with a Learning Effect." *European Journal of Operational Research* 132 (3): 687–693. doi:10.1016/S0377-2217(00)00175-2.

Mosheiov, Gur. 2002. “Complexity Analysis of Job-shop Scheduling with Deteriorating Jobs.” *Discrete Applied Mathematics* 117 (1–3): 195–209. doi:10.1016/S0166-218X(00)00385-1.

Mosheiov, Gur. 2011. “Parallel Machine Scheduling with a Learning Effect.” *The Journal of the Operational Research Society* 52 (10): 1165–1169.

Mosheiov, Gur, and Jeffrey B. Sidney. 2003. “Scheduling with General Job-dependent Learning Curves.” *European Journal of Operational Research* 147 (3): 665–670. doi:10.1016/S0377-2217(02)00358-2.

Ng, C. T., T. C. E. Cheng, A. Bachman, and A. Janiak. 2002. “Three Scheduling Problems with Deteriorating Jobs to Minimize the Total Completion Time.” *Information Processing Letters* 81 (6): 327–333. doi:10.1016/S0020-0190(01)00244-7.

Niu, Qun, Bin Jiao, and Xingsheng Gu. 2008. “Particle Swarm Optimization Combined with Genetic Operators for Job Shop Scheduling Problem with Fuzzy Processing Time.” *Applied Mathematics and Computation* 205 (1): 148–158. doi:10.1016/j.amc.2008.05.086.

Prade, Henri M. 1979. “Using Fuzzy Set Theory in a Scheduling Problem: A Case Study.” *Fuzzy Sets and Systems* 2 (2): 153–165.

Qian, J., and G. Steiner. 2013a. “Fast Algorithms for Scheduling with Learning Effects and Time-dependent Processing times on a Single Machine.” *European Journal of Operational Research* 225 (3): 547–551. doi:10.1016/j.ejor.2012.09.013.

Qian, J., and G. Steiner. 2013b. “Scheduling with Learning Effects and/or Time-dependent Processing times to Minimize the Weighted Number of Tardy Jobs on a Single Machine.” *Mathematical Problems in Engineering* 2013: 1–9. doi:10.1155/2013/972358.

Rostami, Mohammad, Amir Ebrahimzadeh Pilerood, and Mohammad Mahdavi Mazdeh. 2015. “Multi-objective Parallel Machine Scheduling Problem with Job Deterioration and Learning Effect under Fuzzy Environment.” *Computers and Industrial Engineering* 85: 206–215. doi:10.1016/j.cie.2015.03.022.

Rudek, R. 2013. “The Computational Complexity Analysis of the Two-processor Flowshop Problems with Position Dependent Job Processing times.” *Applied Mathematics and Computation* 221: 819–832. doi:10.1016/j.amc.2013.06.086.

Sakawa, Masatoshi, and Ryo Kubota. 2000. “Fuzzy Programming for Multiobjective Job Shop Scheduling with Fuzzy Processing Time and Fuzzy Duedate through Genetic Algorithms.” *European Journal of Operational Research* 120 (2): 393–407. doi:10.1016/S0377-2217(99)00094-6.

Sakawa, Masatoshi, and Tetsuya Mori. 1999. “An Efficient Genetic Algorithm for Job-shop Scheduling Problems with Fuzzy Processing Time and Fuzzy Duedate.” *Computers and Industrial Engineering* 36 (2): 325–341. doi:10.1016/S0360-8352(99)00135-7.

Soroush, H. M. 2016. “Multicriteria Scheduling with Controllable Processing times, Setup times, and Learning, Ageing and Deterioration Effects on a Single Machine.” *European Journal of Industrial Engineering* 10 (4): 455–478. doi:10.1504/EJIE.2016.078140.

Toksarı, M. Duran., 2011. “A Branch and Bound Algorithm for Minimizing Makespan on a Single Machine with Unequal Release Times under Learning Effect and Deteriorating Jobs.” *Computers and Operations Research* 38 (9): 1361–1365. doi:10.1016/j.cor.2010.12.010.

Toksarı, M. Duran, and Ertan Güner. 2008. “Minimizing the Earliness/Tardiness Costs on Parallel Machine with Learning Effects and Deteriorating Jobs: A Mixed Nonlinear Integer Programming Approach.” *The International Journal of Advanced Manufacturing Technology* 38 (7–8): 801–808. doi:10.1007/s00170-007-1128-3.

Toksarı, M. Duran, and Ertan Güner. 2009. “Parallel Machine Earliness/Tardiness Scheduling Problem under the Effects of Position Based Learning and Linear/Nonlinear Deterioration.” *Computers and Operations Research* 36 (8): 2394–2417. doi:10.1016/j.cor.2008.09.012.

Toksari, M. Duran, and Ertan Güner. 2010a. “The Common Due-date Early/Tardy Scheduling Problem on a Parallelmachine under the Effects of Time-dependent Learning and Linear and Nonlinear Deterioration.” *Expert Systems with Applications* 37: 92–112.

Toksari, M. Duran, and Ertan Güner. 2010b. “Parallel Machine Scheduling Problem to Minimize the Earliness/Tardiness Costs with Learning Effect and Deteriorating Jobs.” *Journal of Intelligent Manufacturing* 21 (6): 843–851. doi:10.1007/s10845-009-0260-3.

Tsujimura, Yasuhiro, Seung Hun Park, Inseong Chang, and Mitsuo Gen. 1993. “An Effective Method for Solving Flow Shop Scheduling Problems with Fuzzy Processing times.” *Computers and Industrial Engineering* 25 (1–4): 239–242.

Wang, J.-B. 2006. “A Note on Scheduling Problems with Learning Effect and Deteriorating Jobs.” *International Journal of Systems Science* 37 (12): 827–833. doi:10.1080/00207720600879260.

Wang, Ji-Bo. 2007. “Single-machine Scheduling Problems with the Effects of Learning and Deterioration.” *Omega* 35 (4): 397–402. doi:10.1016/j.omega.2005.07.008.

Wang, J. B., L. Lin, and F. Shan. 2008. “Flow Shop Scheduling with Effects of Learning and Deterioration.” *Journal of Applied Mathematics and Computing* 26 (1–2): 367–379. doi:10.1007/s12190-007-0033-0.

Wang, J.-B., M. Liu, N. Yin, and P. Ji. 2017. “Scheduling Jobs with Controllable Processing Time, Truncated Job-dependent Learning and Deterioration Effects.” *Journal of Industrial and Management Optimization* 13 (2): 1025–1039. doi:10.3934/jimo.2016060.

Wang, Ji Bo, C. T. Ng, and T. C. E. Cheng. 2008. “Single-machine Scheduling with Deteriorating Jobs under a Series–Parallel Graph Constraint.” *Computers and Operations Research* 35 (8): 2684–2693. doi:10.1016/j.cor.2006.12.026.

Wang, Ji Bo, C. T. Daniel Ng, T. C. E. Cheng, and Li Li Liu. 2006. “Minimizing Total Completion Time in a Two-machine Flow Shop with Deteriorating Jobs.” *Applied Mathematics and Computation* 180 (1): 185–193. doi:10.1016/j.amc.2005.11.162.

Wang, J. B., C. T. Ng, T. C. E. Cheng, and L. L. Liu. 2008. “Single-machine Scheduling with a Time-dependent Learning Effect.” *International Journal of Production Economics* 111 (2): 802–811. doi:10.1016/j.ijpe.2007.03.013.

Wang, J. B., and Z. Q. Xia. 2005. “Flow-shop Scheduling with a Learning Effect.” *Journal of the Operational Research Society* 56 (11): 1325–1330. doi:10.1016/j.palgrave.jors.2601856.

Wang, Ji Bo, and Zun Quan Xia. 2006. “Flow Shop Scheduling with Deteriorating Jobs under Dominating Machines.” *Omega* 34 (4): 327–336. doi:10.1016/j.omega.2004.10.006.

Wu, Chin Chia, and Wen Chiung Lee. 2008. “Single-machine Group-scheduling Problems with Deteriorating Setup times and Job-processing times.” *International Journal of Production Economics* 115 (1): 128–133. doi:10.1016/j.ijpe.2008.05.004.

Xie, Yuan, Jianying Xie, and Qinhua Huang. 2005. “Single Machine Scheduling Problem with Fuzzy Due Dates and Fuzzy Precedence.” *Journal of Shanghai University (English Edition)* 9 (5): 450–454.

Yeh, Wei Chang, Peng Jen Lai, Wen Chiung Lee, and Mei Chi Chuang. 2014. “Parallel-machine Scheduling to Minimize Makespan with Fuzzy Processing times and Learning Effects.” *Information Sciences* 269: 142–158. doi:10.1016/j.ins.2013.10.023.

Yin, Yunqiang, Dehua Xu, Kaibiao Sun, and Hongxing Li. 2009. “Some Scheduling Problems with General Position-dependent and Time-dependent Learning Effects.” *Information Sciences* 179 (14): 2416–2425. doi:10.1016/j.ins.2009.02.015.

Yusriski, R., T. M. A. A. Samadhi Sukoyo, and A. H. Halim. 2016. “An Integer Batch Scheduling Model for a Single Machine with Simultaneous Learning and Deterioration Effects to Minimize Total Actual Flow Time.” In *IOP Conference Series: Materials Science and Engineering 2nd International Manufacturing Engineering Conference and 3rd Asia-Pacific Conference onManufacturing Systems (iMEC-APCOMS 2015)*, 12–14 November 2015, Kuala Lumpur, Malaysia, Vol. 114. doi:10.1088/1757-899X/114/1/012073.