Computers and Mathematics with Applications 60 (2010) 1374–1384

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Heuristics for the two-machine flowshop scheduling problem to minimize maximum lateness with bounded processing times

Ali Allahverdi [a,*], Harun Aydilek [b]

[a] Department of Industrial and Management Systems Engineering, College of Engineering and Petroleum, Kuwait University, P.O. Box 5969, Safat, Kuwait
[b] Department of Mathematics and Natural Sciences, Gulf University for Science and Technology, P.O. Box 7207, Hawally 32093, Kuwait

ARTICLE INFO

Article history:
Received 30 November 2009
Received in revised form 9 June 2010
Accepted 11 June 2010

Keywords:
Scheduling
Flowshop
Maximum lateness
Random processing times

ABSTRACT

This study addresses a two-machine flowshop scheduling problem to minimize maximum lateness where processing times are random variables with lower and upper bounds. This problem is NP-hard since the corresponding deterministic problem is known to be NP-hard. Hence, we propose nine heuristics which utilize due dates and the lower and upper bounds on job processing times along with the Earliest Due Date sequence. Furthermore, we propose an algorithm which yields four heuristics. The proposed fourteen heuristics are compared with each other and with a random solution through randomly generated data. Four different distributions (uniform, negative exponential, positive exponential, and normal) of processing times within given lower and upper bounds are investigated. The computational analysis has shown that one of the proposed heuristics performs as the best over all the considered parameters and for the four distributions with an overall average percentage relative error of less than one.

1. Introduction

One assumption made in the vast majority of research on the two-machine flowshop scheduling problem is that job processing times are known fixed values. However, in some real-life scheduling problems, it may not be realistic to assume that job processing times are known in advance as fixed values. Rather, it is more realistic to estimate these times within a lower bound and an upper bound. If the processing time of a job is known with certainty, then the lower and upper bounds of that job will be equal. On the other hand, the gap between the lower and upper bounds of a job will get larger as a job's processing time becomes more uncertain.

In some scheduling environments, it is hard to obtain an exact probability distribution for the job processing time if modeled as a random variable. Thus, a solution obtained by assuming a certain probability distribution may not even be close to the optimal solution for the realization of the process. It has been observed that, although the exact probability distributions of processing times may not be known, upper and lower bounds on processing times are easy to obtain in many practical cases. This information on the bounds of processing times is important, and it can be utilized in finding a solution for the scheduling problem.

For the considered problem, the processing times are random variables with unknown probability distributions when only a lower bound $Lt_{j,m} \geq 0$ and an upper bound $Ut_{j,m} \geq Lt_{j,m}$ of the processing time $t_{j,m}$ of job j ($j \in J = \{1, 2, \ldots, n\}$) on machine m ($m \in M = \{1, 2\}$) are known. Such a flowshop problem can be denoted as $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|\Lambda$; where the first term denotes that it is a two-machine flowshop. The second term indicates that processing times are unknown

* Corresponding author. Fax: +965 24816137.
E-mail addresses: ali.allahverdi@ku.edu.kw (A. Allahverdi), aydilek.h@gust.edu.kw (H. Aydilek).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.06.019

Table 1
Description of some proposed heuristics.

Heuristic sequence	The sequence is obtained by arranging the jobs in non-decreasing order of:
SLL	$d_i – Lt_{i,1} – Lt_{i,2}$
SLU	$d_i – Lt_{i,1} – Ut_{i,2}$
SUL	$d_i – Ut_{i,1} – Lt_{i,2}$
SUU	$d_i – Ut_{i,1} – Ut_{i,2}$
SLA	$d_i – Lt_{i,1} – (Lt_{i,2} + Ut_{i,2})/2$
SUA	$d_i – Ut_{i,1} – (Lt_{i,2} + Ut_{i,2})/2$
SAL	$d_i – (Lt_{i,1} + Ut_{i,1})/2 – Lt_{i,2}$
SAU	$d_i – (Lt_{i,1} + Ut_{i,1})/2 – Ut_{i,2}$
SAA	$d_i – (Lt_{i,1} + Ut_{i,1})/2 – (Lt_{i,2} + Ut_{i,2})/2$

variables with some lower and upper bounds. The last term specifies the performance measure to be minimized. Notice that the problem $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|\Lambda$ can be considered as a stochastic flowshop problem under *uncertainty* of processing times when there is no prior information about the probability distributions of the random processing times. In this case, it is only known that the processing time of each job will fall between some given lower and upper bounds with probability one. Uncertainty modeling is important for some environments, in particular, for supply chains, e.g., Barve et al. [1], Brun et al. [2], Hornung and Monch [3]. Some researchers have recently proposed the use of a fuzzy set theory to model the uncertainty, e.g., [4,5].

Allahverdi and Sotskov [6] addressed the problem $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|C_{\max}$, where $C_{\max}$ denotes makespan. They proposed some dominance relations to reduce the set of the search space in finding the optimal solution. Matsveichuk et al. [7] also addressed the same problem, and they provided sufficient conditions for obtaining a dominant solution out of the reduced set attained by using earlier established dominance relations. For the same problem, Allahverdi and Aydilek [8] proposed different heuristics using the lower and upper bounds, and compared the heuristics based on randomly generated data. Allahverdi and Aydilek [8] reported that three of their proposed heuristics perform well with an overall average error of less than one percent. Moreover, for symmetric distributions, Allahverdi and Aydilek [8] showed that one of the heuristics, which applies Johnson's algorithm to the average of the lower and upper bounds, performs as the best, with an overall average percentage error of less than one. Allahverdi et al. [9] proposed some dominance relations for the same problem where setup times are considered as separate from processing times.

For the $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|TCT$ problem, Sotskov et al. [10] provided some dominance relations, where *TCT* denotes the total completion time. The performance measure of the total completion time is very important as it is directly related to the cost of inventory. The significance of minimizing the total cost of inventory has been discussed by many researchers, e.g., [11–17]. Aydilek and Allahverdi [18] also considered the problem $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|TCT$, and proposed eleven heuristics which utilize the lower and upper bounds on job processing times. Aydilek and Allahverdi [18] reported that the heuristics using the information on the bounds of job processing times on both machines perform much better than those using the information on one of the two machines.

Allahverdi [19] provided some dominance relations for the problem of $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|L_{\max}$, where $L_{\max}$ denotes maximum lateness. These dominance relations help reduce the solution set of the problem and, for some restricted problems, the size of solution set may be small. In particular, when the lower and upper bounds are very close to each other, then the size of the solution set can be small. Nevertheless, in general, it may be impossible to reduce the solution set by these dominance relations to a small number. In this paper, we present different heuristics for the problem that can be used to obtain a good solution regardless of the closeness of the lower and upper bounds.

2. Heuristics

The two-machine flowshop scheduling problem with the objective of minimizing maximum lateness is known to be NP-hard [20] even when the lower and upper bounds are the same for all the jobs, i.e., the deterministic problem. Therefore, the problem of $F2|Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}|L_{\max}$ is also NP-Complete. This means that it is highly unlikely to find an optimal solution with polynomial time complexity. Therefore, in this section we propose different heuristics to solve the problem.

It is well known that the sequence Earliest Due Date (*EDD*) minimizes $L_{\max}$ for the deterministic single machine problem. It is not, in general, optimal for a two-machine flowshop. However, it can be used as an approximate solution for our problem.

It is also well known that not only the size of due dates is important but also job processing times on both the first and the second machines are important. Hence, the job processing times have to be taken into account. We define the sequence *SLL* to be the sequence obtained by arranging the jobs in non-decreasing order of $d_i – Lt_{i,1} – Lt_{i,2}$. Similarly, other sequences are defined in Table 1.

In addition to the above simple heuristic sequences, next, we propose an algorithm called ADD.

Let A denote a matrix of size n by 4, where n denotes the number of jobs. The first column of matrix A, $A(:, 1)$, denotes the lower bounds of job processing times on machine 1 while the second column, $A(:, 2)$, denotes the upper bounds of job processing times on machine 1. Similarly, the third and the fourth columns $(A(:, 3), A(:, 4))$ of matrix A denote the lower and upper bounds of processing times on machine 2, respectively.

Algorithm DD (ADD)

Step 0: Define $AL = [A(:,1)A(:,3)]$

Step 1: Let $\sigma_1 = \{\text{all jobs}\}$, $ADL = \phi$

Step 2: Choose job $i \in \sigma_1$ with the minimum of d_i
Remove this job from σ_1 and place it in the first position of *ADL*. Set $k = 2$

Step 3: If $k = n$, Go to Step 5,
Place each job $i \in \sigma_1$ in position k of *ADL* one at a time
Compute $L_{\max}$ of the partial sequence *ADL* for each by using the matrix *AL* for job processing times.
Remove the job yielding the maximum $L_{\max}$ from σ_1 and place it in position k of *ADL*. In case of ties, choose the first one.
Let $k := k + 1$.

Step 4: Go to Step 3.

Step 5: Assign the last remaining job in σ_1 to the last position of *ADL*.

Now, by using the same Steps 0–5 find the two sequences *ADU* and *ADA* for which the job processing times are $AU = [A(:,2)A(:,4)]$ and $AA = (AL + AU)/2$, respectively.

Step 6: Now find $L_{\max}$ for the sequences *ADL*, *ADU*, and *ADA* by using the job processing times matrix *AA*. The sequence among the sequences *ADL*, *ADU*, and *ADA* and *EDD* with the minimum $L_{\max}$ is the *ADD* sequence.

3. Computational experiments

The proposed heuristics *SLL*, *SLU*, *SUL*, *SUU*, *SLA*, *SUA*, *SAL*, *SAU*, *SAA*, *ADL*, *ADA*, *ADU* and *ADD* along with *EDD* are evaluated based on randomly generated data following different distributions. We compared the performance of the heuristics using three measures: average percentage relative error (Error), standard deviation (Std) out of five hundred replicates, and the percentage of times the best solution is obtained (Count).

The percentage relative error is defined as $100 * (L_{\max}$ of the heuristic $- L_{\max}$ of the best heuristic out of 14 heuristics$)/(L_{\max}$ of the worst heuristic out of 14 heuristics $- L_{\max}$ of the best heuristic out of 14 heuristics). Based on this definition of the error, the error of the worst heuristic yields a value of 100 while that of the best heuristic gives a value of zero.

It should be noted that we initially included a random sequence to measure the performance of all the heuristics, i.e., 15 rather than 14 in the above definition of the error. It has been observed that the average error of the random sequence was more than 90% compared to those of other heuristics while the average error of the worst heuristic among the 14 heuristics was less than 20%. Therefore, even the performance of the worst heuristic was significantly better than a random solution. Since the inclusion of the random solution made the comparison of the best performing heuristics difficult, we have removed the random solution. Hence, all the results reported in this paper are based on comparing the 14 heuristics with each other.

The upper bounds of processing times are generated from uniform distributions such that $Ut_{i,j} \in U(1, 100)$. The lower bounds $LBt_{i,j}$ on processing times are generated from $LBt_{i,j} = UBt_{i,j} - \Delta$, where Δ was randomly generated from a uniform distribution from two different ranges, namely, $\Delta \in U(0, 10)$, and $\Delta \in U(0, 20)$. Once the lower and upper bounds for each job have been generated, then an instance (a realization) for job processing times is generated following different distributions. We consider the uniform, exponential (negative and positive), and normal distributions. For the normal distribution, the lower and upper bounds were set to the lower and upper bounds of the processing times, and not to negative and positive infinities as in an ordinary normal distribution. That is, the lower and upper bounds were truncated, and hence, whenever a number below the lower bound or above the upper bound was generated, the number was repeated until a number between the two bounds was obtained. It should be noted that the probability of a number being generated outside the range is extremely small. The descriptions of the normal and exponential distributions are given in the Appendix. These distributions are more or less representative of many distributions, since the extreme cases are considered.

The job due dates have been randomly uniformly generated in the range $[CP(1 - T - R/2), CP(1 - T + R/2)]$, where *CP* denotes the time at which all the jobs are expected to be completed. The parameter T denotes the tardiness factor while R stands for relative range of due dates. This is the standard method used in the scheduling literature to generate due dates for computational experiments, e.g., [21]. For the single machine problem, *CP* denotes the sum of processing times for all jobs. For the multiple-machine problem, however, a lower bound for the makespan is usually used as a *CP* value. For our problem, first Johnson's algorithm is used to compute makespan ($C_{\max}(L)$) using $Lt_{i,1}$ and $Lt_{i,2}$ for the job processing times on machines 1 and 2, respectively. Secondly, Johnson's algorithm is used to compute makespan ($C_{\max}(U)$) using $Ut_{i,1}$ and $Ut_{i,2}$ for the job processing times on the first and the second machines, respectively. Then CP was set to the average of $C_{\max}(L)$ and $C_{\max}(U)$. The values of T and R are usually taken to be between 0 and 1 in the literature. Due to a large number of parameters, 0.4 and 0.6 are chosen as values for T while 0.2 and 0.6 are chosen as values for R, which results in four combinations of T and R.

The total number of cases is 160 as five different values of jobs (20, 40, 60, 80, 100), four different distributions (uniform, negative exponential, positive exponential, normal), two different values of $\Delta(U(0, 10), U(0, 20))$, two different values of T (0.4, 0.6) and two different values of R (0.2, 0.6) are considered. For each case, 500 replicates (realizations or instances) are generated to evaluate the performance of the proposed heuristics. These replicates are repeated 100 times. This results in a total of 8,000,000 problems. It should be noted that a much larger number of replicates (up to 2000) has been tested and it was found that 500 replicates were good enough to have a very small standard deviation.

Table 2
Average error and count of heuristics for the uniform distribution ($\Delta \in U(0, 10)$).

	$N = 20$		$N = 40$		$N = 60$		$N = 80$		$N = 100$	
	Error	Count	Error	Count	Error	Count	Error	Count	Error	Count
$T = 0.4$ $R = 0.2$										
EDD	43.52	34	48.75	26	40.70	33	43.77	33	46.49	25
SLL	74.91	4	74.04	5	71.22	6	77.34	1	71.18	8
SLU	76.40	6	70.47	5	73.66	6	77.76	3	74.54	4
SUL	73.27	8	72.94	7	73.08	5	75.71	2	74.00	5
SUU	74.44	7	71.64	10	74.65	5	77.48	2	74.59	3
SLA	76.65	5	73.57	5	71.27	5	76.99	3	73.95	6
SUA	74.26	8	73.51	9	73.13	5	76.81	2	72.00	5
SAL	73.42	6	73.77	6	72.70	6	77.76	2	72.62	6
SAU	75.48	7	71.42	8	74.03	5	78.19	3	72.68	4
SAA	75.54	7	71.00	9	70.52	4	77.22	3	73.11	6
ADL	26.04	44	19.19	50	27.70	41	23.75	44	19.64	47
ADA	25.65	42	19.56	47	24.69	51	23.15	41	16.36	55
ADU	26.59	42	23.43	49	28.07	42	22.30	45	17.12	52
ADD	2.02	91	2.87	84	3.22	88	2.14	91	2.57	88
$T = 0.4$ $R = 0.6$										
EDD	32.43	47	31.53	47	24.76	47	30.34	56	29.11	54
SLL	66.18	15	70.63	13	71.30	10	71.12	6	76.02	10
SLU	66.34	17	69.47	13	73.08	9	69.10	8	73.37	12
SUL	66.28	15	70.61	13	75.09	10	72.47	8	73.60	13
SUU	65.98	17	70.72	13	72.32	12	71.71	7	73.43	11
SLA	64.49	16	68.52	13	74.23	8	69.34	7	73.52	9
SUA	65.05	17	71.58	13	72.39	12	73.15	8	73.52	11
SAL	67.85	15	70.44	13	74.58	11	70.86	7	74.61	10
SAU	65.20	16	69.47	13	74.56	10	70.60	8	73.13	12
SAA	65.69	15	69.60	13	73.92	10	69.74	7	75.46	10
ADL	35.45	40	39.35	35	37.72	37	38.96	42	37.01	27
ADA	34.62	40	34.92	41	39.00	35	39.88	38	38.29	29
ADU	31.99	43	33.75	43	37.79	43	43.12	35	38.11	28
ADD	4.83	83	5.15	88	4.14	86	2.36	97	4.74	79
$T = 0.6$ $R = 0.2$										
EDD	38.21	40	43.70	31	33.67	39	46.76	25	45.88	25
SLL	72.83	10	69.78	10	76.10	4	80.08	0	79.42	1
SLU	72.34	9	65.51	13	76.07	4	76.59	2	80.26	1
SUL	74.11	10	70.06	10	74.24	6	80.77	1	76.31	4
SUU	74.28	7	70.61	10	75.43	7	78.09	1	77.98	1
SLA	73.45	8	66.50	13	76.58	5	78.10	2	78.27	2
SUA	72.75	9	69.56	9	75.33	7	80.46	1	74.77	1
SAL	72.87	8	70.57	13	75.39	6	80.41	2	76.64	3
SAU	74.29	8	68.68	12	74.63	6	77.89	1	79.27	3
SAA	74.47	8	66.08	13	74.94	6	77.99	1	77.29	3
ADL	26.09	42	22.28	47	22.56	43	20.62	50	22.52	41
ADA	26.80	42	23.89	41	28.13	35	20.40	49	20.01	43
ADU	21.25	52	23.51	49	27.93	35	19.84	52	17.62	54
ADD	3.05	90	3.11	87	2.10	90	1.59	93	2.29	87
$T = 0.6$ $R = 0.6$										
EDD	23.01	60	27.62	51	25.38	51	22.01	60	26.80	60
SLL	73.84	13	76.28	10	75.74	5	77.04	9	77.80	9
SLU	70.55	14	70.63	12	76.71	8	76.66	9	79.71	8
SUL	71.79	13	74.17	12	75.47	8	77.07	7	74.95	7
SUU	71.20	14	70.04	13	76.56	8	79.79	6	80.02	6
SLA	71.26	14	73.27	11	75.23	7	77.94	9	79.46	8
SUA	71.76	14	72.71	12	75.10	8	77.97	7	77.62	6
SAL	70.82	14	76.13	10	75.15	6	75.99	9	77.42	8
SAU	71.07	12	72.28	12	74.82	9	76.24	8	79.96	6
SAA	71.66	13	72.73	12	75.95	7	77.50	8	77.57	7
ADL	35.45	41	33.06	44	32.15	36	42.17	30	42.39	30
ADA	37.16	38	32.10	42	33.18	38	41.19	33	43.95	28
ADU	39.40	41	30.88	46	32.23	36	39.80	28	41.12	28
ADD	3.96	91	4.29	88	1.45	92	2.75	89	2.52	93

The computational results (Error and Count) for the proposed heuristics are given in Table 2 ($\Delta \in U(0, 10)$) and Table 3 ($\Delta \in U(0, 20)$) for the case of the uniform distribution. The Std values were very small (around one percent of the average error) and due to space limitation they are not reported. The results for the average errors are summarized in Figs. 1 and 2.

Table 3
Average error and count of heuristics for the uniform distribution ($\Delta \in U(0, 20)$).

	$N = 20$		$N = 40$		$N = 60$		$N = 80$		$N = 100$	
	Error	Count	Error	Count	Error	Count	Error	Count	Error	Count
$T = 0.4$ $R = 0.2$										
EDD	39.41	33	42.05	32	49.61	19	44.77	18	48.26	22
SLL	65.79	6	66.67	2	71.53	4	69.56	2	69.39	7
SLU	68.05	8	70.22	5	72.33	5	67.52	3	70.86	5
SUL	71.33	6	68.12	2	74.04	4	71.64	2	67.61	7
SUU	70.76	5	72.96	1	76.63	5	73.17	2	72.23	5
SLA	66.41	8	66.56	1	71.76	3	68.36	2	73.17	6
SUA	72.74	5	70.90	0	72.75	3	71.40	2	70.50	5
SAL	66.58	7	66.48	2	72.66	4	69.26	1	65.28	6
SAU	68.92	6	70.56	1	74.61	5	69.72	2	68.89	6
SAA	71.60	6	67.68	0	72.79	2	69.27	2	71.06	5
ADL	25.79	37	25.01	34	22.65	43	19.96	45	28.19	30
ADA	26.58	31	27.61	31	25.72	36	22.74	40	24.90	35
ADU	27.67	28	30.10	34	25.61	34	24.72	38	22.32	37
ADD	6.19	84	7.54	73	3.47	82	4.33	84	4.19	82
$T = 0.4$ $R = 0.6$										
EDD	22.99	53	23.82	52	25.30	52	26.70	47	26.21	48
SLL	72.55	6	65.77	8	70.37	7	69.39	3	65.05	9
SLU	71.63	7	73.06	9	74.65	4	68.80	6	67.42	11
SUL	75.10	7	66.18	10	66.30	5	68.47	4	69.82	6
SUU	69.78	8	71.43	9	70.96	5	66.77	7	69.38	9
SLA	71.18	7	71.52	8	71.85	6	65.68	4	66.02	9
SUA	72.83	4	68.88	9	70.67	5	65.88	5	70.35	6
SAL	75.14	5	68.02	8	66.45	7	67.49	2	66.50	7
SAU	70.02	7	73.86	7	71.37	6	66.77	6	68.81	11
SAA	72.41	5	68.36	8	70.72	5	65.66	5	68.47	5
ADL	40.99	27	42.50	24	41.06	27	39.36	33	44.01	25
ADA	40.47	31	42.69	23	39.82	20	35.14	27	43.45	23
ADU	43.11	34	42.60	26	39.44	19	35.67	23	42.77	18
ADD	6.60	79	4.75	86	6.21	80	4.81	81	5.73	75
$T = 0.6$ $R = 0.2$										
EDD	32.40	46	46.71	18	44.22	25	46.89	23	47.87	21
SLL	65.96	7	69.37	5	73.85	2	71.70	3	74.03	1
SLU	66.72	10	70.21	3	72.68	4	69.33	4	76.17	2
SUL	66.43	6	72.48	4	71.98	5	74.30	2	71.71	1
SUU	71.75	7	73.60	6	72.95	2	72.73	3	74.92	1
SLA	64.68	9	67.59	4	73.45	2	69.71	5	74.37	2
SUA	70.49	6	74.13	3	71.76	6	70.78	2	73.97	0
SAL	63.73	5	72.04	4	73.03	5	71.67	3	72.35	0
SAU	69.23	9	70.88	4	72.36	4	68.94	3	78.47	1
SAA	65.32	8	73.01	3	72.53	4	69.44	4	73.96	0
ADL	30.82	37	25.57	32	21.69	32	18.26	42	20.97	40
ADA	32.06	38	26.49	34	23.18	29	20.19	33	20.97	29
ADU	32.15	33	26.01	42	25.33	34	21.05	30	21.05	35
ADD	3.29	89	6.93	75	3.67	77	2.38	83	2.31	85
$T = 0.6$ $R = 0.6$										
EDD	24.57	56	28.00	51	23.67	44	28.11	47	28.21	50
SLL	73.96	7	67.69	5	66.75	10	68.82	4	67.04	6
SLU	72.44	11	65.68	4	68.77	10	68.37	3	65.89	9
SUL	71.57	10	69.52	7	69.10	10	71.42	8	68.06	5
SUU	74.63	12	66.09	6	70.02	8	73.73	4	70.03	8
SLA	73.13	9	64.21	4	67.51	9	69.12	3	65.63	8
SUA	73.84	10	70.86	3	71.43	8	74.09	4	67.12	5
SAL	71.66	8	66.96	7	68.16	9	70.49	4	65.50	6
SAU	74.21	13	65.40	4	70.95	9	73.08	3	67.47	5
SAA	74.86	9	63.15	4	68.10	9	73.00	4	65.29	8
ADL	40.24	31	39.88	27	37.65	34	35.43	26	36.29	27
ADA	43.43	28	41.56	25	37.69	35	35.91	33	35.01	29
ADU	45.06	27	43.13	21	39.28	36	36.49	30	34.51	33
ADD	5.18	84	6.48	78	4.97	79	6.61	82	4.91	82

The results for the other distributions are only summarized in Figs. 3–8 and the corresponding tables are omitted due to space limitation.

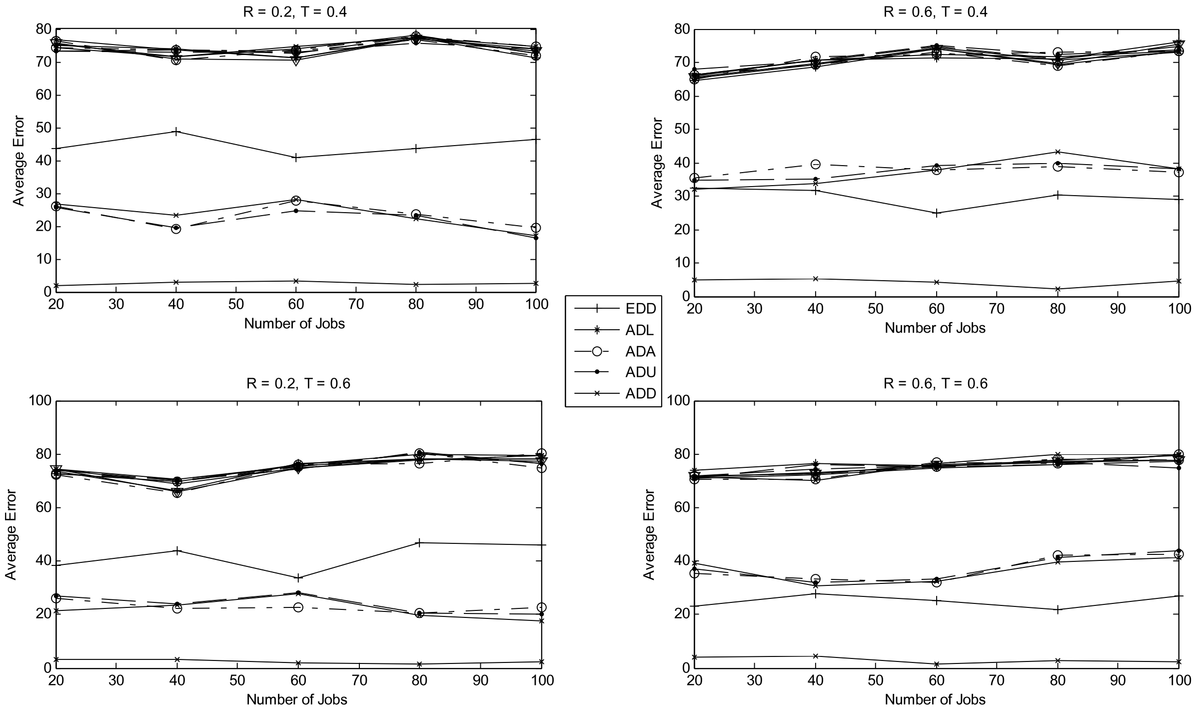

Fig. 1. Average error of heuristics for uniform distribution ($\Delta \in U(0, 10)$).

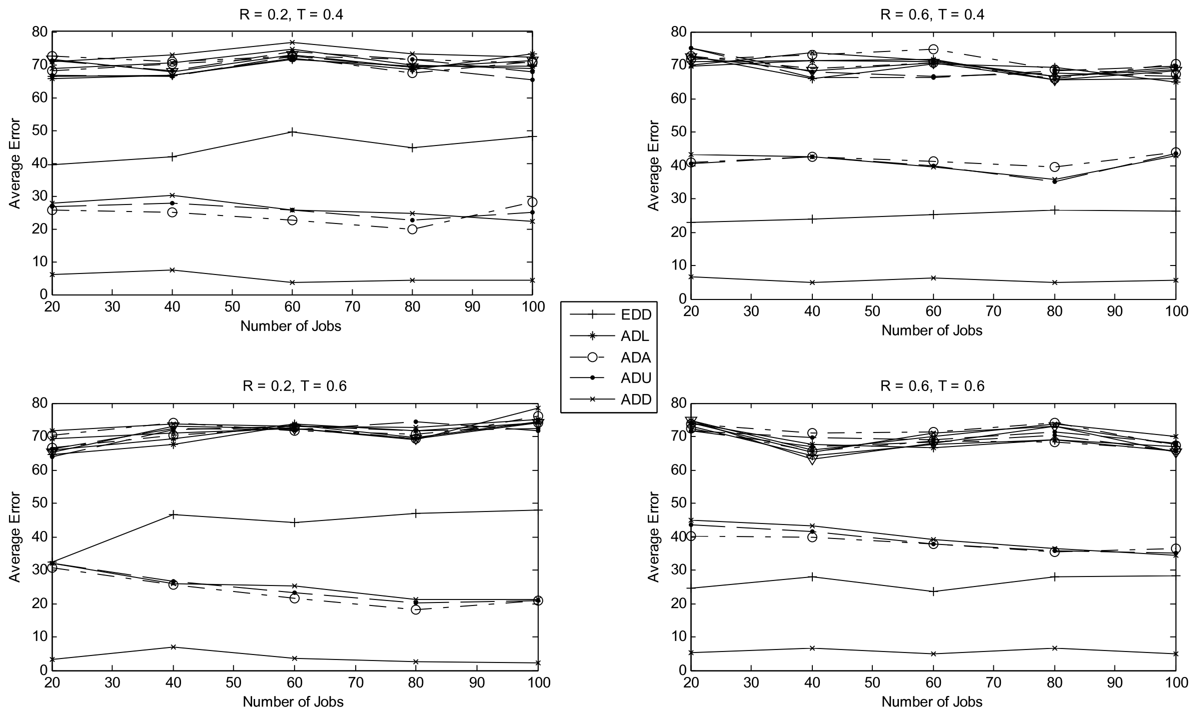

Fig. 2. Average error of heuristics for uniform distribution ($\Delta \in U(0, 20)$).

As can be seen from Figs. 1 and 2, the heuristics based on the information of the lower and upper bounds on both machines, i.e., *SLL*, *SLU*, *SUL*, *SUU*, *SLA*, *SUA*, *SAL*, *SAU*, and *SAA*, perform very poorly compared to the heuristics of *EDD*, *ADL*, *ADA*, *ADU*, and *ADD*. The average errors of *SLL*, *SLU*, *SUL*, *SUU*, *SLA*, *SUA*, *SAL*, *SAU*, and *SAA* are very close to each other, clustered around 70%, for all combinations of Δ, R, and T. Due to space limitation, the legends of *SLL*, *SLU*, *SUL*, *SUU*, *SLA*, *SUA*, *SAL*, *SAU*, and *SAA* are omitted in the graphs. The heuristics *ADL*, *ADA*, *ADU* perform better

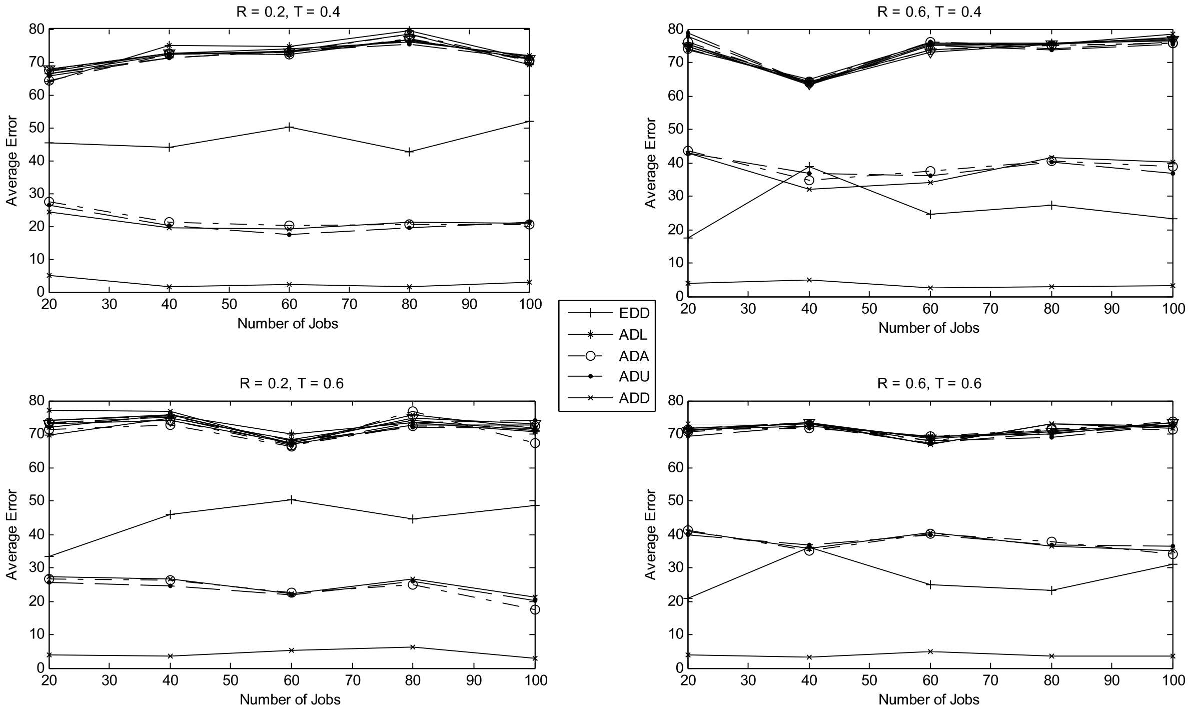

Fig. 3. Average error of heuristics for negative exponential distribution ($\Delta \in U(0, 10)$).

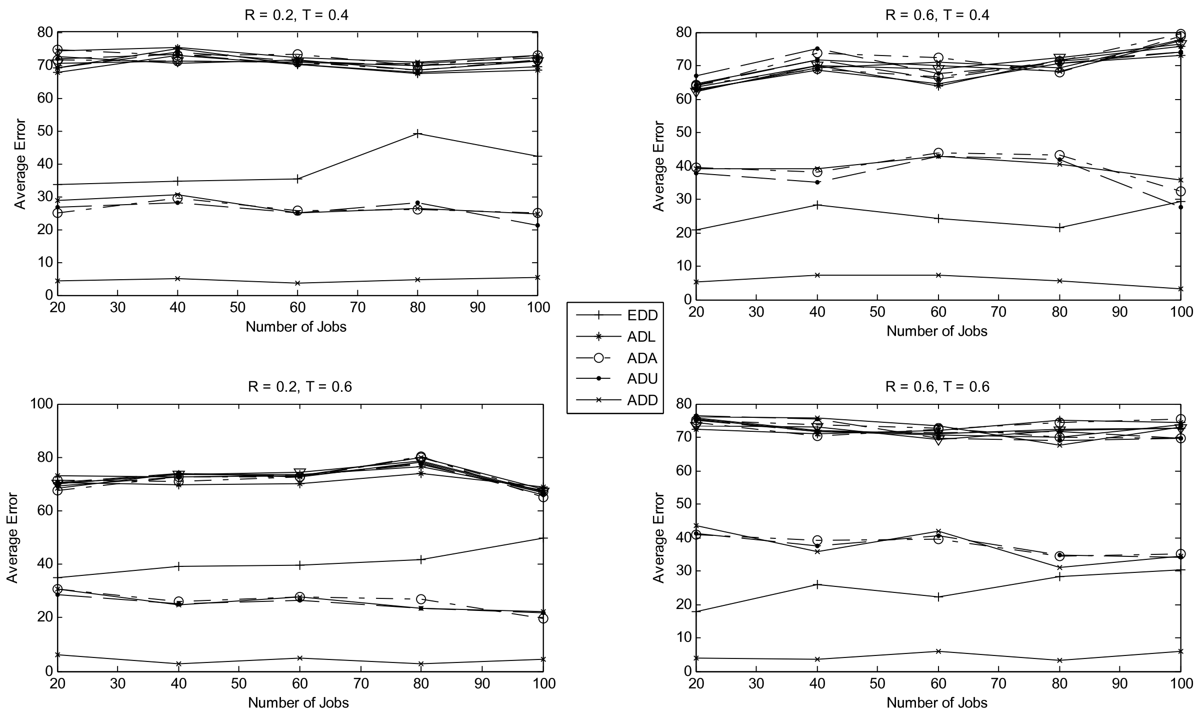

Fig. 4. Average error of heuristics for negative exponential distribution ($\Delta \in U(0, 20)$).

than *EDD* for tight due dates (when $R = 0.2$) while *EDD* performs better than *ADL*, *ADA*, *ADU* for loose due dates (when $R = 0.6$). Among the heuristics *EDD*, *ADL*, *ADA*, *ADU*, *ADD*, the heuristic *ADD* performs much better than the remaining four heuristics for all combinations of Δ, R, and T. As a result, the proposed heuristic *ADD* is superior to all other heuristics for all combinations.

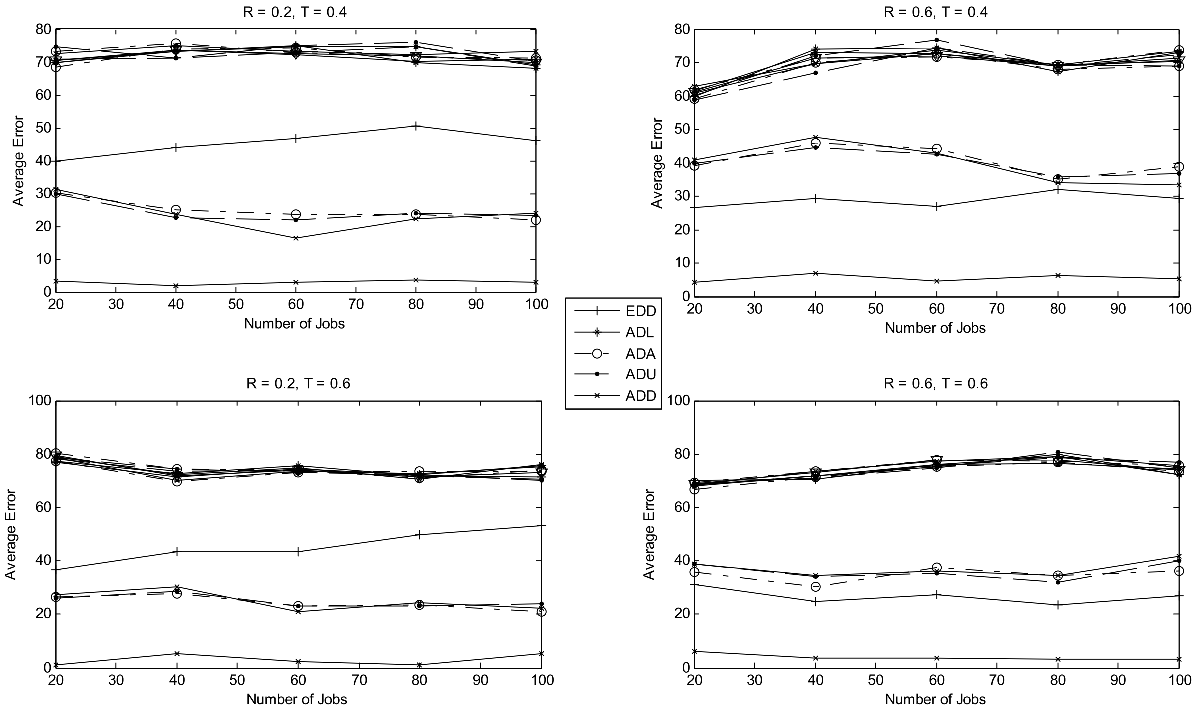

Fig. 5. Average error of heuristics for positive exponential distribution ($\Delta \in U(0, 10)$).

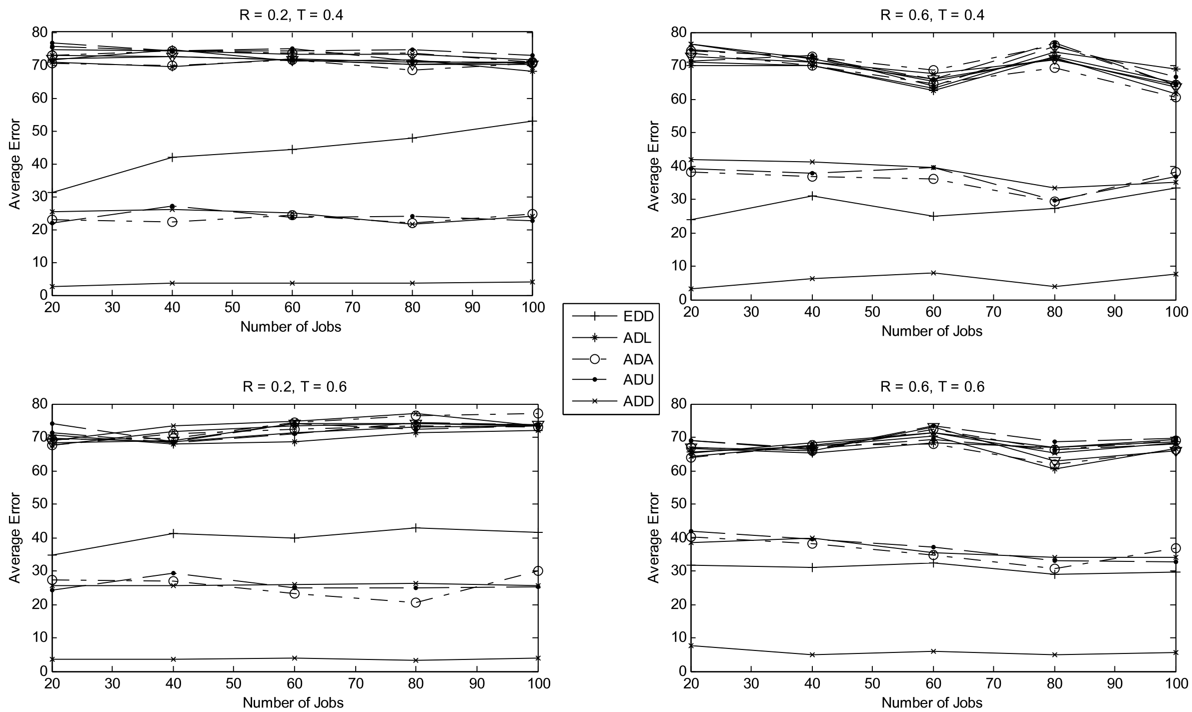

Fig. 6. Average error of heuristics for positive exponential distribution ($\Delta \in U(0, 20)$).

Figs. 3–6 summarize the average errors for all the heuristics for the negative and positive exponential distributions, respectively. The results for the normal distribution are summarized in Figs. 7–8. In general, the average errors for these three distributions are similar to those of the uniform distribution. Hence, the conclusions for the uniform distribution are valid for these three distributions as well.

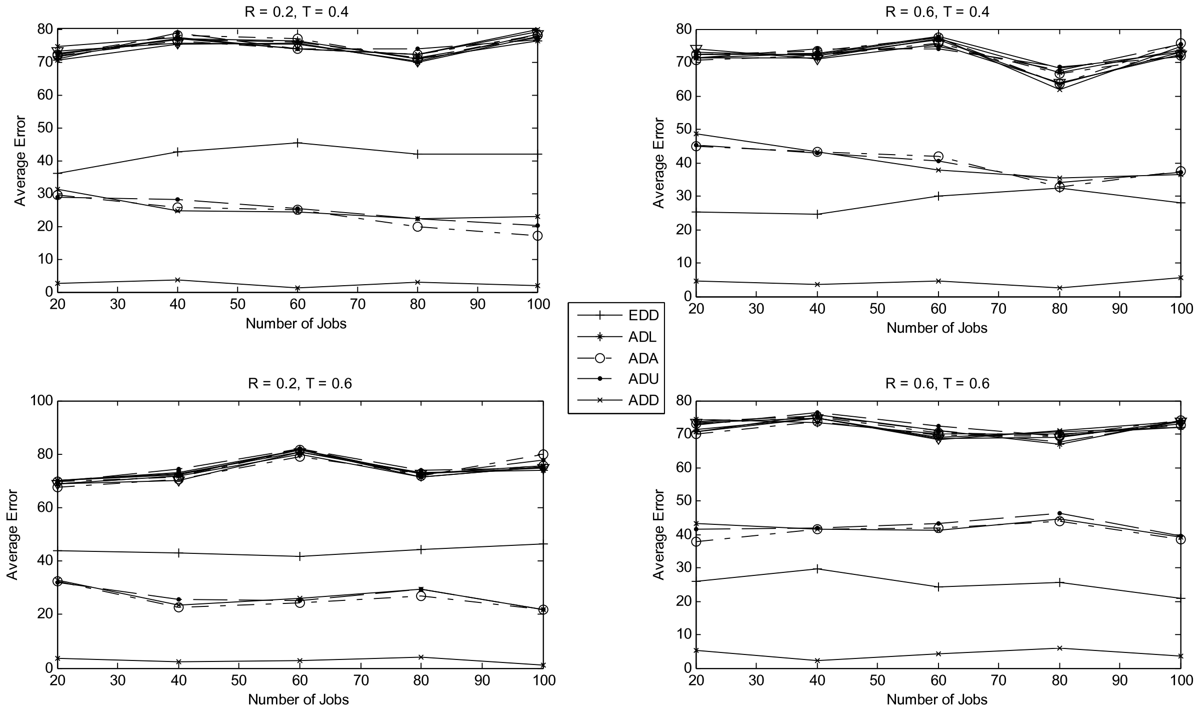

Fig. 7. Average error of heuristics for normal distribution ($\Delta \in U(0, 10)$).

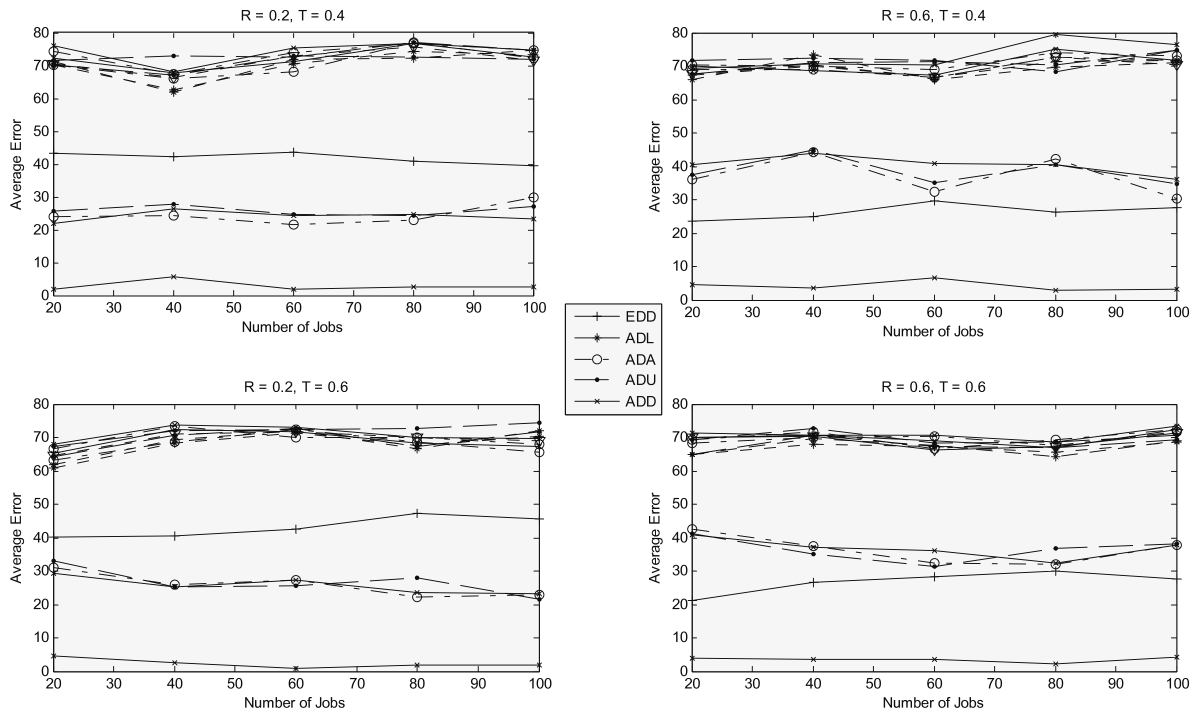

Fig. 8. Average error of heuristics for normal distribution ($\Delta \in U(0, 20)$).

In summary, the overall average errors of *EDD*, *ADL*, *ADA*, *ADU*, and *ADD* for all combinations of Δ, R, and T and distributions, are 35.04, 31.21, 31.45, 31.66, and 3.94 respectively. As reported earlier, the standard deviations were very small compared to the average errors.

As reported earlier, the performances of heuristics reported so far are compared with each other for the 14 considered heuristics. However, when a random solution was also included in the solution set (comparing 15 solutions), then, the overall average relative error of *ADD* was 0.7. Hence, it can be stated safely that the heuristic *ADD* performs very well. Due to space limitation, the results when comparing 15 heuristics are not reported in the paper.

4. Conclusions

We addressed the two machine flowshop scheduling problem to minimize maximum lateness, where processing times are modeled as general random variables, i.e., distribution free. The only known information is the lower and upper bounds for the processing times of each job. We presented different heuristics, and compared the performance of these heuristics through extensive computational experiments. The computational experiments indicated that one of the heuristics performed well with an overall relative error of 0.7.

The importance of setup times has been addressed by Allahverdi et al. [22,23]. In this paper, setup times are ignored or assumed to be included in the processing times. This assumption is valid for some scheduling environments. However, the assumption may not be valid for some other scheduling environments, e.g., see [24]. Therefore, another possible extension is to consider the problem addressed in this paper with setup times.

Scheduling problems with random and bounded processing times have been addressed in the flowshop and jobshop environments, but not in single or parallel machine environments or flexible flowshops, e.g., [25–28]. Therefore, single or parallel machine problems can be addressed with random and bounded processing times.

Appendix

In this Appendix, we describe the Exponential (negative and positive) and Normal distributions that have been used in Section 3 to evaluate the performances of the proposed heuristics. The Exponential and Normal distributions that have been considered in this paper are truncated, the definitions of which are given in this Appendix.

1. *Exponential distribution*

The *pdf* for the truncated exponential distribution is $f(x) = \frac{\alpha e^{\lambda x}}{e^{\alpha Ut_{i,j}} - e^{\alpha Lt_{i,j}}}$ for $x \in (Lt_{i,j}, Ut_{i,j})$ and zero otherwise. α is taken as 0.1 for positive exponential, and -0.1 for negative exponential.

2. *Normal distribution*

We considered the truncated normal distribution with a mean of $\mu = \frac{Lt_{i,j}+Ut_{i,j}}{2}$ and a standard deviation of $\sigma = \frac{Ut_{i,j}-Lt_{i,j}}{6}$.

References

[1] A. Barve, A. Kanda, R. Shankar, The role of human factors in agile supply chains, European Journal of Industrial Engineering 3 (2009) 2–20.
[2] A. Brun, K.F. Salama, M. Gerosa, Selecting performance measurement systems: matching a supply chain's requirements, European Journal of Industrial Engineering 3 (2009) 336–362.
[3] A. Hornung, L. Monch, Heuristic approaches for determining minimum cost delivery quantities in supply chains, European Journal of Industrial Engineering 2 (2008) 377–400.
[4] T. Chen, Estimating and incorporating the effects of a future QE project into a semiconductor yield learning model with a fuzzy set approach, European Journal of Industrial Engineering 3 (2009) 207–226.
[5] R. Parameshwaran, P.S.S. Srinivasan, M. Punniyamoorthy, S.T. Charunyanath, C. Ashwin, Integrating fuzzy analytical hierarchy process and data envelopment analysis for performance management in automobile repair shops, European Journal of Industrial Engineering 3 (2009) 450–467.
[6] A. Allahverdi, Y.N. Sotskov, Two-machine flowshop minimum length scheduling problem with random and bounded processing times, International Transactions in Operational Research 10 (2003) 65–76.
[7] N.M. Matsveichuk, Y.N. Sotskov, N.G. Egorova, T.C. Lai, Schedule execution for two-machine flow-shop with interval processing times, Mathematical and Computer Modelling 49 (2009) 991–1011.
[8] A. Allahverdi, H. Aydilek, Heuristics for two-machine flowshop scheduling problem to minimize makespan with bounded processing times, International Journal of Production Research (2009) (in press).
[9] A. Allahverdi, T. Aldowaisan, Y.N. Sotskov, Two-machine flowshop scheduling problem to minimize makespan or total completion time with random and bounded setup times, International Journal of Mathematics and Mathematical Sciences 39 (2003) 2475–2486.
[10] Y.N. Sotskov, A. Allahverdi, T.C. Lai, Flowshop scheduling problem to minimize total completion time with random and bounded processing times, Journal of Operational Research Society 55 (2004) 277–286.
[11] O.P. Singh, S. Chand, Supply chain design in the electronics industry incorporating JIT purchasing, European Journal of Industrial Engineering 3 (2009) 21–44.
[12] R.D.H. Warburton, EOQ extensions exploiting the Lambert W function, European Journal of Industrial Engineering 3 (2009) 45–69.
[13] S. Sharma, On price increases and temporary price reductions with partial backordering, European Journal of Industrial Engineering 3 (2009) 70–89.
[14] S. Mohan, G. Mohan, A. Chandrasekhar, Multi-item, economic order quantity model with permissible delay in payments and a budget constraint, European Journal of Industrial Engineering 2 (2008) 446–460.
[15] K. Ramaekers, G.K. Janssens, On the choice of a demand distribution for inventory management models, European Journal of Industrial Engineering 2 (2008) 479–491.
[16] J. Liu, L. Wu, Z. Zhou, A time-varying lot size method for the economic lot scheduling problem with shelf life considerations, European Journal of Industrial Engineering 2 (2008) 337–355.
[17] E. Hassini, R. Vickson, Lot-splitting for inspection in a synchronized two-stage manufacturing process with finite production rates and random out-of-control shifts in the first stage, European Journal of Industrial Engineering 2 (2008) 207–229.
[18] H. Aydilek, A. Allahverdi, Two-machine flowshop scheduling problem with bounded processing times to minimize total completion time, Computers and Mathematics with Applications 59 (2010) 684–693.
[19] A. Allahverdi, Two-machine flowshop scheduling problem to minimize maximum lateness with bounded setup and processing times, Kuwait Journal of Science and Engineering 33 (2006) 233–251.
[20] J.K. Lenstra, A.H.G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, Annals of Discrete Mathematics 1 (1977) 343–362.
[21] Y.D. Kim, A new branch and bound algorithm for minimizing mean tardiness in two-machine flowshops, Computers and Operations Research 20 (1993) 391–401.

[22] A. Allahverdi, J.N.D. Gupta, T. Aldowaisan, A review of scheduling research involving setup considerations, OMEGA. The International Journal of Management Sciences 27 (1999) 219–239.

[23] A. Allahverdi, C.T. Ng, T.C.E. Cheng, M.Y. Kovalyov, A survey of scheduling problems with setup times or costs, European Journal of Operational Research 187 (2008) 985–1032.

[24] V. Vinod, R. Sridharan, Simulation-based metamodels for scheduling a dynamic job shop with sequence-dependent setup times, International Journal of Production Research 47 (2009) 1425–1447.

[25] S.A. Fahmy, T.Y. ElMekkawy, S. Balakrishnan, Deadlock-free scheduling of flexible job shops with limited capacity buffers, European Journal of Industrial Engineering 2 (2008) 231–252.

[26] C. Pessan, J.L. Bouquard, E. Neron, An unrelated parallel machines model for an industrial production resetting problem, European Journal of Industrial Engineering 2 (2008) 153–171.

[27] P. Damodaran, N.S. Hirani, M.C. Velez-Gallego, Scheduling identical parallel batch processing machines to minimize makespan using genetic algorithms, European Journal of Industrial Engineering 3 (2009) 187–206.

[28] Y. Chen, X. Li, R. Sawhney, Restricted job completion time variance minimisation on identical parallel machines, European Journal of Industrial Engineering 3 (2009) 261–276.