Computers and Mathematics with Applications 59 (2010) 684–693

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Two-machine flowshop scheduling problem with bounded processing times to minimize total completion time

Harun Aydilek [a,*], Ali Allahverdi [b,1]

[a] Department of Natural Sciences and Mathematics, Gulf University for Science and Technology, P.O. Box 7207, Hawally 32093, Kuwait

[b] Department of Industrial and Management Systems Engineering, College of Engineering and Petroleum, Kuwait University, P.O. Box 5969, Safat, Kuwait

ARTICLE INFO

Article history:
Received 14 April 2009
Received in revised form 10 October 2009
Accepted 19 October 2009

Keywords:
Scheduling
Flowshop
Total completion time
Random and bounded processing times

ABSTRACT

We consider the two-machine flowshop scheduling problem where jobs have random processing times which are bounded within certain intervals. The objective is to minimize total completion time of all jobs. The decision of finding a solution for the problem has to be made based on the lower and upper bounds on job processing times since this is the only information available. The problem is NP-hard since the special case when the lower and upper bounds are equal, i.e., the deterministic case, is known to be NP-hard. Therefore, a reasonable approach is to come up with well performing heuristics. We propose eleven heuristics which utilize the lower and upper bounds on job processing times based on the Shortest Processing Time (SPT) rule. The proposed heuristics are compared through randomly generated data. The computational analysis has shown 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. It has also shown that one of the proposed heuristics performs as the best for different distributions with an overall average percentage error of less than one.

1. Introduction

The deterministic (where processing times are known with certainty) two-machine flowshop scheduling problem with the minimization of total completion time has been considered by many researchers for a long time. The performance measure of 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., [1–7].

The deterministic two-machine flowshop scheduling problem is unary NP-hard, e.g. see [8]. Therefore, the main research has been focused on the development of either implicit enumeration techniques or heuristics. For some scheduling environments, it is perfectly valid to assume that job processing times are deterministic in which case the implicit enumeration techniques and heuristics appeared in the literature can be utilized. Nonetheless, for some other scheduling environments, the assumption of deterministic processing times may not be applicable. As stated by Soroush [9,10], the random variation in processing times needs to be taken into account while searching for a solution.

The flowshop scheduling problem has been addressed by some researchers where job processing times follow certain probability distributions. For example, Cunningham and Dutta [11] and Ku and Niu [12] addressed the problem where jobs have exponentially distributed processing times while Kalczynski and Kamburowski [13] addressed the problem for the case where job processing times follow Weibull distribution. Portougal and Trietsch [14] suggested that variance

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

1 Fax: +965 24816137.

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.10.025

Table 1
Upper and lower bounds of job processing times.

Job j	1	2	3
$Lt_{j,1}$	4	5	10
$Ut_{j,1}$	10	9	18
$Lt_{j,2}$	2	8	3
$Ut_{j,2}$	6	20	9

reduction should be also taken into account while selecting a solution for stochastic flowshops. Portougal and Trietsch [15] concluded that both the mean and the variance are required for valid comparison of different schedules. Assuming that job processing times are random variables with known cumulative distribution functions, Portougal and Trietsch [16] developed and evaluated two heuristics. For some scheduling environments, even when job processing times are deterministic, job completion times may be stochastic as a result of random machine breakdowns, e.g., [17]. Moreover, some researchers have recently proposed the use of a fuzzy set theory to model the uncertainty in scheduling problems, e.g., [18–20]. Uncertainly modeling is very important for some environment, in particular, for supply chains, e.g., [21–23].

For some scheduling environments, it is hard to obtain exact probability distributions for random processing times, and therefore assuming a specific probability distribution is not realistic. Usually, solutions obtained after assuming a certain probability distribution are not even close to the optimal solution. It has been observed that, although the exact probability distribution of job processing times may not be known, upper and lower bounds on job processing times are easy to obtain in many cases. Hence, this information on the bounds of job processing times should be utilized in finding a solution for the scheduling problem. The scheduling problem with bounded processing times was first introduced by Lai et al. [24], and later studied by different researchers including Lai and Sotskov [25], Allahverdi and Sotskov [26], and Matsveichuk et al. [27]. It should be noted that this situation may occur for jobs that are processed for the first time so that not much information is available. Otherwise, the average or most likely value of processing times could be taken into account as being used in project scheduling, e.g., [28].

For the two-machine flowshop scheduling problem with bounded processing times to minimize total completion time, Sotskov et al. [29] and Allahverdi [30] provided some dominance relations. These dominance relations help in reducing the solution set of the problem, and for some restricted problems, the size of solution set may be very 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 that can be used to obtain a good solution regardless of the closeness of the lower and upper bounds.

The paper is organized as follows. In Section 2, the problem is defined. Section 3 presents the proposed heuristics. The computational evaluation of these heuristics is conducted in Section 4 and concluding remarks are given in Section 5.

2. Problem definition

The two-machine flowshop scheduling problem is considered with the objective of minimizing total completion time. There are n jobs ready to be processed by two machines where each job first has to be processed by the first machine, and then it has to be processed by the second machine. It should be noted that the considered objective function is equivalent to minimizing average job completion time.

It is well known that permutation schedules are dominant for the deterministic two-machine flowshop problem of minimizing mean or total completion time criterion. That is, one only needs to consider the same sequence of jobs on both machines in order to find the optimal schedule. Permutation schedules are also known to be dominant for the problem of two-machine flowshop with random processing times. Since we assume that the processing times are random variables with given lower and upper bounds, permutation schedules are dominant for the problem under consideration as well.

Let $Lt_{j,m}$ and $Ut_{j,m}$ denote the lower bound and the upper bound of processing time of job j on machine m, respectively. The exact value $t_{j,m}$ of the processing time of job j on machine m is not known until machine m completes processing the job j. However, it is known that the processing time will be somewhere between its lower and upper bounds. In other words, $Lt_{j,m} \leq t_{j,m} \leq Ut_{j,m}$. Even if $Lt_{j,m} = t_{j,m} = Ut_{j,m}$ for all jobs and both machines, it is known that the problem is NP-hard. For the problem that we consider in this paper, the exact realizations of job processing times are not known. Therefore, the quality of an earlier developed solution (i.e., a sequence) might change depending on the exact realization of job processing times, which will only be known after all jobs have been processed on both machines. For example, consider a problem of three jobs where the lower and upper bounds on processing times are given in Table 1.

Since job processing times can be any real number between lower and upper bounds, there is an infinite number of realizations. For example, five different realizations for the problem described in Table 1 are given in Table 2. Unfortunately, we do not know which one of these or others (an infinite number of realizations) will occur. Therefore, it is almost impossible to find a sequence that will remain optimal for all realizations. Of course, after jobs have been completed we know the exact values of $t_{i,j}$, where we are in a position to find the optimal sequence, for at least small number of jobs since we can find the solution by exhaustive enumeration. Unfortunately, we have to make a decision about a sequence before the schedule takes place, which we can only use the lower and upper bounds on processing times, i.e., based on the data given in Table 1.

Table 2
Five different realizations for the example given in Table 1.

Realization		Job 1	Job 2	Job 3
1	$t_{j,1}$	4	5	10
	$t_{j,2}$	2	8	3
2	$t_{j,1}$	10	9	18
	$t_{j,2}$	6	20	9
3	$t_{j,1}$	7	7	14
	$t_{j,2}$	4	14	6
4	$t_{j,1}$	9	6	14
	$t_{j,2}$	3	18	4
5	$t_{j,1}$	10	5	10
	$t_{j,2}$	6	8	4

It can be shown that the sequence $(1, 2, 3)$ is optimal for the realizations 1, 2, 3, and 4. However, the sequence $(1, 2, 3)$ is not optimal for realization 5. The optimal sequence for realization 5 is either the sequence $(2, 1, 3)$ or $(2, 3, 1)$. In fact, an example can easily be constructed such that each of the five realizations in Table 2 may have a different optimal solution.

It should be noted that the only information available before making a decision on a solution is the knowledge of lower and upper bounds on job processing times. Therefore, these bounds should be used in finding a solution for the problem. In the following section, we propose different heuristics which utilize the information about these bounds.

3. Proposed heuristics

It is known that ordering jobs based on Shortest Processing Time (SPT) minimizes total completion time for the deterministic single machine scheduling problem. For the two-machine flowshop scheduling problem, when $Ut_{i,j} = Lt_{i,j}$ for all $i = 1, 2, \ldots, n$ and $j = 1, 2$, the problem reduces to the deterministic two-machine flowshop scheduling problem. Therefore, two SPT rules (SPT1 and SPT2) can be defined. Allahverdi and Tatari [31] defined SPT1 as the SPT based on job processing times on the first machine while SPT2 as the SPT based on job processing times on the second machine. It should be noted that even the deterministic version of our problem does not have a polynomial solution as it is known that the problem is NP-hard, e.g., [8]. Moreover, $Ut_{i,j} \neq Lt_{i,j}$ for at least some of the jobs, and the exact realization of $t_{i,j}$ will not be known unless job i has finished its processing on machine j. However, a decision on when to process job i on machine j has to be made earlier, i.e., before the realization of $t_{i,j}$. In other words, a decision can be made only on the available data on job i on machine j, which are the lower and upper bounds, i.e., $Ut_{i,j}$ and $Lt_{i,j}$. It should be also noted that it does not make sense to use meta-heuristics since the exact values of $t_{i,j}$ are not known. In other words, there is no point in spending so much time as to where job i should be placed since even small changes in the processing times would significantly affect the quality of schedule obtained before realization of processing times. Therefore, the only possible option would be to use heuristics which utilize $Ut_{i,j}$ and $Lt_{i,j}$.

We can use the idea of SPT while searching for heuristics. However, the exact values of $t_{i,j}$ are not known while the lower and upper bounds of $t_{i,j}$, i.e., $Ut_{i,j}$ and $Lt_{i,j}$ are known. Hence, we can use the idea of SPT by using $Ut_{i,j}$ and $Lt_{i,j}$ values. For example, a sequence can be obtained by ordering the jobs according to SPT based on $Lt_{i,1}$, i.e., based on the lower bounds on machine 1. We call this sequence as SPTL1. Similarly, another sequence can be obtained by ordering jobs according to SPT based on Ut_{i2}. This sequence is called SPTU2. By using $Ut_{i,j}$ and $Lt_{i,j}$ values, nine other sequences can be obtained. Table 3 lists all the proposed sequences. The heuristic SPTA1 is obtained by ordering jobs following SPT according to the average of the lower and upper bounds on job processing times on machine 1 while SPTA2 is obtained by doing the same for the second machine. The heuristics SPTLL, SPTUU, SPTLU, SPTUL, and SPTAA are obtained by taking into account the information on job processing times on both machines. For example, the heuristic SPTLU is obtained by following SPT according to the average of $Lt_{i,1}$ and $Ut_{i,2}$. The rest of the heuristics are described in Table 3.

The heuristic SPTL1 for the problem given in Table 1 results in the sequence $(1, 2, 3)$. This is because $Lt_{1,1} = 4$, $Lt_{2,1} = 5$, $Lt_{3,1} = 10$. Therefore, first the first job is to be processed, then the second job, and finally the third job. Similarly, the heuristic SPTU1 gives the sequence $(2, 1, 3)$ since among $Ut_{i,1}$'s, the smallest processing time is 9 (job 2), the next smallest is 10 (job 1), and finally is 18 (job 3). The sequence of jobs obtained by all heuristics are given in Table 4. It should be noted that since there are only three jobs for the given problem, most of the heuristics result in the same solution. For large size problems, however, the probability of having several heuristics resulting in the same solution is very small.

4. Computational experiments

The proposed heuristics SPTL1, SPTUI, SPTA1, SPTL2, SPTU2, SPTA2, SPTLL, SPTUU, SPTLU, SPTUL, and SPTAA are evaluated based on randomly generated data following different distributions. We compared the performance of the heuristics using two measures: average percentage relative error (Error) and standard deviation (Std) out of two thousand replicates. The

Table 3
Description of the proposed eleven heuristics.

Heuristic name	Order the jobs based on SPT according to
SPTL1	$Lt_{i,1}$
SPTU1	$Ut_{i,1}$
SPTA1	$(Lt_{i,1+}Ut_{i,1})/2$
SPTL2	$Lt_{i,2}$
SPTU2	$Ut_{i,2}$
SPTA2	$(Lt_{i,2+}Ut_{i,2})/2$
SPTLL	$(Lt_{i,1+}Lt_{i,2})/2$
SPTUU	$(Ut_{i,1+}Ut_{i,2})/2$
SPTLU	$(Lt_{i,1+}Ut_{i,2})/2$
SPTUL	$(Ut_{i,1+}Lt_{i,2})/2$
SPTAA	$[(Lt_{i,1+}Ut_{i,1})/2 + (Lt_{i,2+}Ut_{i,2})/2]/2$

Table 4
Heuristic solution for the problem given in Table 1.

Heuristic name	Sequence of the jobs
SPTL1	1, 2, 3
SPTU1	2, 1, 3
SPTA1	1, 2, 3
SPTL2	1, 3, 2
SPTU2	1, 3, 2
SPTA2	1, 3, 2
SPTLL	1, 2, 3
SPTUU	1, 3, 2
SPTLU	1, 3, 2
SPTUL	1, 2, 3
SPTAA	1, 3, 2

percentage error is defined as 100* (*total completion time of the heuristic – total completion time of the best heuristic out of* 11 *heuristics*)/(*total completion time of the best heuristic out of* 11 *heuristics*).

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 uniform distribution from five different ranges, namely, $\Delta \in U(0, 5)$, $\Delta \in U(0, 10)$, $\Delta \in U(0, 15)$, $\Delta \in U(0, 20)$, and $\Delta \in U(0, 40)$. 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 distributions of uniform, exponential (negative and positive), and normal. 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 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 were obtained. It should be noted that the probability of a number being generated outside the range is extremely small. The detail descriptions of the normal and exponential distributions are given in the Appendix. These distributions are more or less representative to many distributions since the extreme cases are considered.

The total number of cases is 100 as five different values of jobs (40, 60, 80, 100, 200), four different distributions (uniform, positive exponential, negative exponential, normal), and five different values of $\Delta(U(0, 5), U(0, 10), U(0, 15), U(0, 20), U(0, 40))$ are considered. For each case, 2000 replicates (realizations or instances) are generated to evaluate the performance of the proposed heuristics. This results in a total of 200,000 problems. It should be noted that a much larger number of replicates (up to 10,000) has been tested and it was found that 2000 replicates were good enough to have a very small standard deviation as indicated in Table 5.

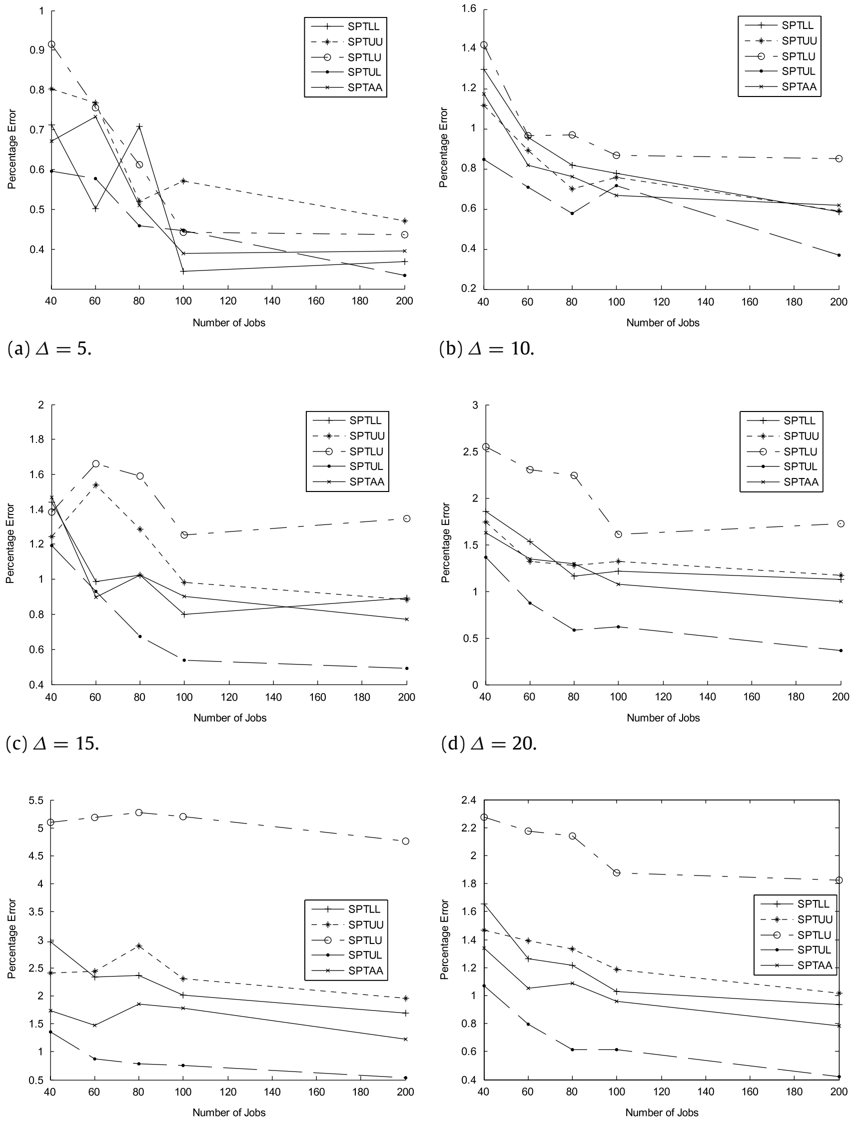
The computational results for the proposed heuristics are given in Table 5 for the case of uniform distribution. As can be seen from the table, the heuristics based on the information of either the lower or upper bound on only one machine, i.e., SPTL1, SPTU1, SPTA1, SPTL2, SPTU2, SPTA2, perform very poorly compared to the heuristics based on the information on both machines, i.e., SPTLL, SPTUU, SPTLU, SPTUL, and SPTAA. The performances of the heuristics SPTL1, SPTU1, SPTA1, SPTL2, SPTU2, SPTA2 were also very poor compared to those of SPTLL, SPTUU, SPTLU, SPTUL, and SPTAA for the normal and exponential (both negative and positive) distributions. This result is expected since the heuristics SPTL1, SPTU1, SPTA1, SPTL2, SPTU2, and SPTA2 take into account the information on a single machine while the heuristics SPTLL, SPTUU, SPTLU, SPTUL, and SPTAA are obtained by considering the information on both machines. Therefore, the results of heuristics SPTL1, SPTU1, SPTA1, SPTL2, SPTU2, and SPTA2 will not be compared for the rest of the analysis. This will also make it easier to compare the rest of well performing heuristics. The results of Table 5 are summarized in Fig. 1 for the well performing heuristics. Moreover, for the sake of brevity, only the summary results are given in Figs. 2–4 for other distributions. It should be noted that the standard deviations (Std), out of two thousand replicates, were significantly small. Moreover, comparison

Table 5
Computational results for uniform distribution.

		$n = 40$		$n = 60$		$n = 80$		$n = 100$		$n = 200$	
		Error	Std	Error	Std	Error	Std	Error	Std	Error	Std
$\Delta = 5$	SPTL1	13.40	0.33	14.91	0.29	15.39	0.26	18.24	0.23	21.34	0.19
	SPTU1	13.41	0.33	14.86	0.29	15.38	0.26	18.24	0.23	21.31	0.19
	SPTA1	13.42	0.33	14.89	0.29	15.39	0.26	18.24	0.23	21.33	0.19
	SPTL2	28.36	0.29	30.14	0.23	30.22	0.20	31.53	0.21	33.76	0.14
	SPTU2	28.41	0.29	30.12	0.23	30.24	0.20	31.45	0.21	33.72	0.14
	SPTA2	28.45	0.29	30.16	0.23	30.18	0.20	31.45	0.21	33.72	0.14
	SPTLL	0.71	0.05	0.50	0.05	0.71	0.05	0.34	0.03	0.37	0.03
	SPTUU	0.80	0.05	0.77	0.05	0.52	0.04	0.57	0.04	0.47	0.03
	SPTLU	0.92	0.06	0.76	0.05	0.61	0.04	0.44	0.04	0.44	0.03
	SPTUL	0.59	0.05	0.58	0.04	0.46	0.04	0.45	0.03	0.33	0.03
	SPTAA	0.67	0.05	0.73	0.05	0.51	0.04	0.39	0.04	0.40	0.03
$\Delta = 10$	SPTL1	15.16	0.65	16.87	0.57	19.17	0.53	20.29	0.48	23.71	0.38
	SPTU1	15.31	0.65	16.67	0.56	19.12	0.53	20.18	0.49	23.72	0.38
	SPTA1	15.18	0.65	16.74	0.56	19.07	0.53	20.19	0.48	23.72	0.38
	SPTL2	29.54	0.60	31.19	0.50	32.14	0.44	32.55	0.41	36.97	0.30
	SPTU2	29.52	0.59	31.17	0.50	32.09	0.44	32.57	0.41	36.82	0.30
	SPTA2	29.51	0.60	31.20	0.50	32.14	0.44	32.50	0.41	36.84	0.30
	SPTLL	1.30	0.16	0.96	0.11	0.82	0.12	0.78	0.11	0.59	0.07
	SPTUU	1.12	0.16	0.89	0.12	0.70	0.10	0.76	0.11	0.59	0.08
	SPTLU	1.42	0.19	0.97	0.13	0.97	0.11	0.87	0.12	0.85	0.09
	SPTUL	0.85	0.14	0.71	0.10	0.58	0.09	0.72	0.09	0.37	0.06
	SPTAA	1.17	0.16	0.82	0.12	0.76	0.09	0.67	0.10	0.62	0.08
$\Delta = 15$	SPTL1	17.35	1.04	16.59	0.88	18.11	0.79	21.61	0.70	23.54	0.58
	SPTU1	17.24	1.05	16.33	0.88	17.93	0.79	21.45	0.70	23.44	0.58
	SPTA1	17.30	1.05	16.38	0.88	17.95	0.79	21.46	0.70	23.43	0.58
	SPTL2	30.69	0.87	31.01	0.72	32.40	0.65	33.90	0.65	35.73	0.43
	SPTU2	30.18	0.87	31.40	0.73	32.08	0.64	33.96	0.64	35.69	0.43
	SPTA2	30.48	0.86	31.21	0.72	32.19	0.64	33.98	0.64	35.74	0.44
	SPTLL	1.44	0.28	0.98	0.22	1.02	0.20	0.80	0.19	0.89	0.16
	SPTUU	1.24	0.27	1.54	0.24	1.29	0.20	0.98	0.18	0.89	0.14
	SPTLU	1.38	0.32	1.66	0.24	1.59	0.22	1.25	0.23	1.35	0.18
	SPTUL	1.19	0.23	0.93	0.17	0.67	0.15	0.54	0.13	0.49	0.10
	SPTAA	1.47	0.28	0.90	0.20	1.02	0.18	0.90	0.18	0.77	0.13
$\Delta = 20$	SPTL1	18.14	1.30	16.52	1.16	19.85	1.07	20.41	0.97	24.21	0.77
	SPTU1	17.51	1.29	16.47	1.16	19.60	1.07	20.23	0.97	24.25	0.77
	SPTA1	17.65	1.29	16.37	1.16	19.54	1.07	20.20	0.97	24.20	0.77
	SPTL2	33.80	1.21	33.74	0.93	34.74	0.86	35.09	0.79	37.59	0.59
	SPTU2	34.15	1.20	33.52	0.91	34.63	0.88	35.09	0.78	37.69	0.59
	SPTA2	34.18	1.20	33.64	0.91	34.58	0.88	35.19	0.79	37.75	0.59
	SPTLL	1.85	0.46	1.53	0.29	1.16	0.28	1.22	0.24	1.13	0.21
	SPTUU	1.74	0.42	1.32	0.31	1.27	0.32	1.32	0.27	1.17	0.20
	SPTLU	2.55	0.47	2.31	0.39	2.24	0.39	1.61	0.29	1.72	0.26
	SPTUL	1.36	0.42	0.88	0.22	0.58	0.17	0.62	0.16	0.37	0.11
	SPTAA	1.63	0.40	1.35	0.30	1.29	0.29	1.07	0.22	0.89	0.20
$\Delta = 40$	SPTL1	18.14	1.30	16.52	1.16	19.85	1.07	20.41	0.97	24.21	0.77
	SPTU1	17.51	1.29	16.47	1.16	19.60	1.07	20.23	0.97	24.25	0.77
	SPTA1	17.65	1.29	16.37	1.16	19.54	1.07	20.20	0.97	24.20	0.77
	SPTL2	33.80	1.21	33.74	0.93	34.74	0.86	35.09	0.79	37.59	0.59
	SPTU2	34.15	1.20	33.52	0.91	34.63	0.88	35.09	0.78	37.69	0.59
	SPTA2	34.18	1.20	33.64	0.91	34.58	0.88	35.19	0.79	37.75	0.59
	SPTLL	1.85	0.46	1.53	0.29	1.16	0.28	1.22	0.24	1.13	0.21
	SPTUU	1.74	0.42	1.32	0.31	1.27	0.32	1.32	0.27	1.17	0.20
	SPTLU	2.55	0.47	2.31	0.39	2.24	0.39	1.61	0.29	1.72	0.26
	SPTUL	1.36	0.42	0.88	0.22	0.58	0.17	0.62	0.16	0.37	0.11
	SPTAA	1.63	0.40	1.35	0.30	1.29	0.29	1.07	0.22	0.89	0.20

of heuristics based on Std were almost the same as the comparison based on the average percentage errors. Therefore, for the sake of brevity, the results for Std will not be reported and comparison will be made only on the percentage errors.

From Figs. 1–4, it can be seen that the heuristics SPTUL and SPTAA, in general, perform better than the other three heuristics of SPTLL, SPTUU, and SPTLU. The good performance of SPTAA is not surprising since the sequence of jobs is determined based on the average of both lower and upper bounds of job processing times on both machines. Of the five considered heuristics (SPTLL, SPTUU, SPTLU, SPTUL, SPTAA), SPTUL is the best performing heuristic, in general, for all the



(a) $\Delta = 5$.
(b) $\Delta = 10$.
(c) $\Delta = 15$.
(d) $\Delta = 20$.
(e) $\Delta = 40$.
(f) Average over Δ.

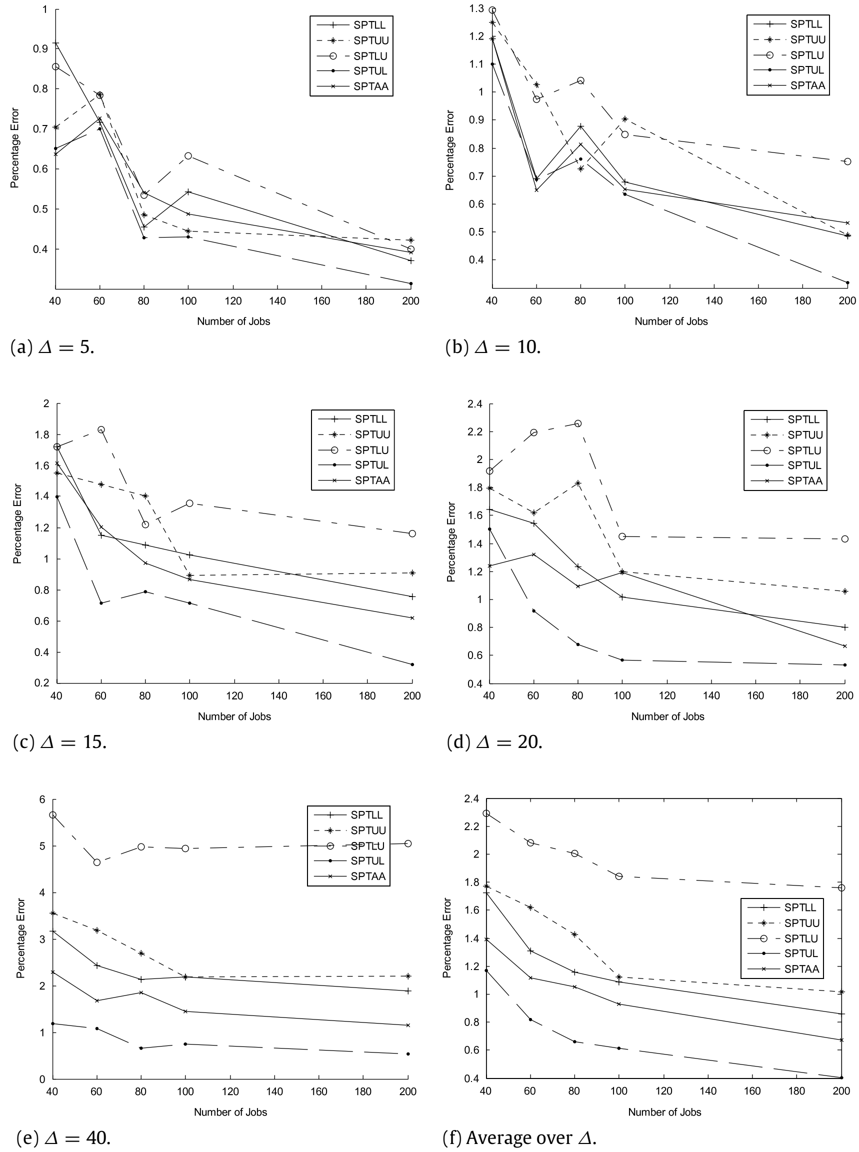
Fig. 1. Average percentage error for uniform distribution.

considered distributions. The overall average percentage error of SPTUL is less than one percent. Since this is the best performing heuristic for all distributions, it can be safely used in finding out a solution for the problem. The second best heuristic is SPTAA for all distributions except for negative exponential distribution for which SPTLL is next best heuristic. This is not surprising since it is more likely that processing times will be closer to the lower bounds.

The difference between the performance of the heuristics of SPTLL, SPTUU, SPTLU, SPTUL, SPTAA gets larger as Δ gets larger. This is expected since as Δ approaches to zero, then the lower and upper bounds of job processing times approach to one another in which case all heuristics will yield the same solution. On the other hand, when Δ is large, then the difference between the lower and upper bounds will be large, and hence, each of the heuristic will give different solution in which case heuristic performances will be far from each other.

Figs. 1–4 indicate that the performance of heuristics does not change much as the number of jobs, n, changes. Even though it seems that for n up to 100, the percentage errors of heuristics seems to be decreasing but it should be noted that the errors are relative errors and not the absolute errors. In general, the differences between the percentage of errors of the considered heuristics do not change much. Hence, it can be concluded that the number of jobs does not affect the performance of the proposed heuristics.

In summary, the heuristics taking into account the lower and upper bounds of job processing times on both machines perform much better than those which take into account the lower and upper bounds on one machine only. Furthermore, among those taking into account both bounds on both machines, SPTUL performs as the best heuristic with an overall percentage error of less than one.



(a) $\Delta = 5$. (b) $\Delta = 10$. (c) $\Delta = 15$. (d) $\Delta = 20$. (e) $\Delta = 40$. (f) Average over Δ.

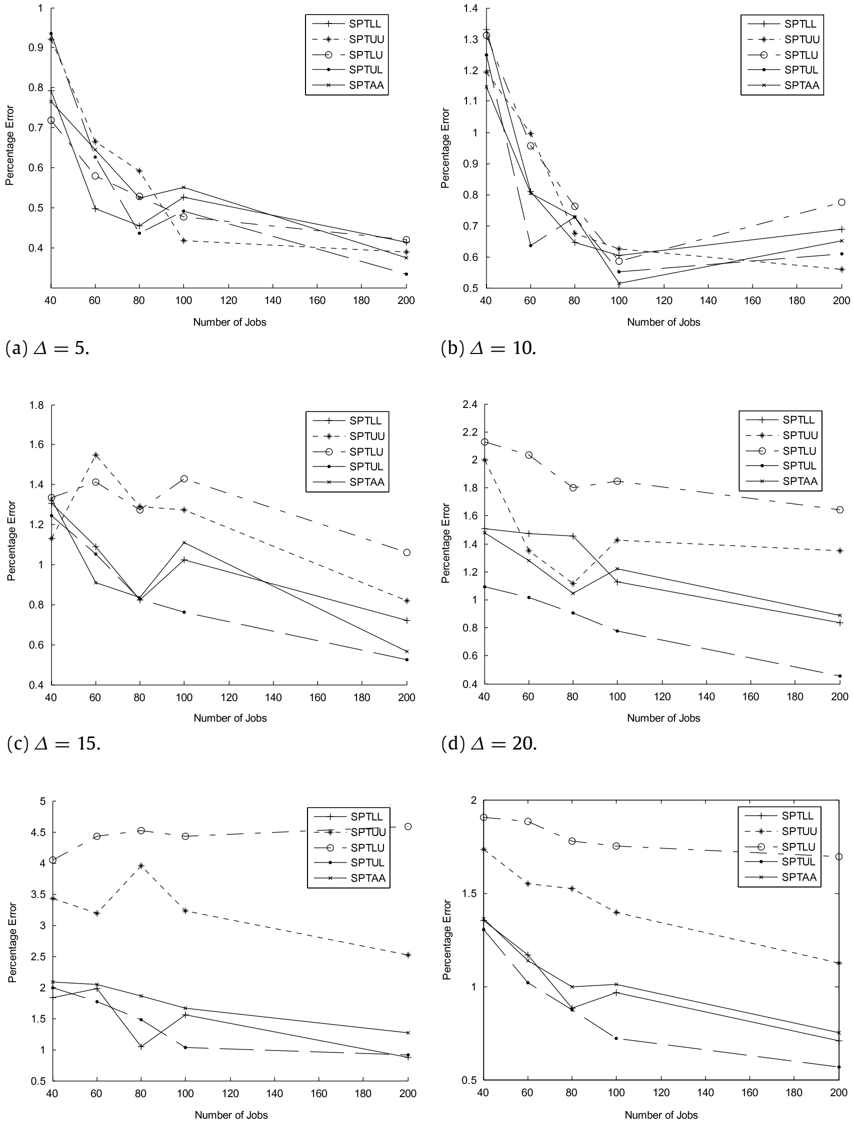
Fig. 2. Average percentage error for normal distribution.

5. Conclusion

The two-machine flowshop scheduling problem to minimize total completion time was addressed. Processing times were modeled as random variables with generic distributions, i.e., no specific distributions were assumed. The only known information about processing times is the lower and upper bounds. Given the deterministic version of the problem is NP-hard, different heuristics were proposed, where the heuristics are constructed by taking into account the lower and upper bounds of job processing times since this is the only known information. The performance of the heuristics was evaluated through an extensive computational experimentation. The computational experiments indicate 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. It has also shown that one of the proposed heuristics performs as the best for different distributions with an overall average percentage error of less than one.

There are different possible extensions to the problem addressed in this paper. One possible extension is to address the problem with respect to maximum lateness criterion. Another possible extension would be to consider no-idle flowshops, e.g., [32].

The importance of setup times has been addressed by Allahverdi et al. [33,34]. 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 [35]. Therefore, another possible extension



(a) $\Delta = 5$.

(b) $\Delta = 10$.

(c) $\Delta = 15$.

(d) $\Delta = 20$.

(e) $\Delta = 40$.

(f) Average over Δ.

Fig. 3. Average percentage error for negative exponential distribution.

is to consider the problem addressed in this paper with setup times. Moreover, it is assumed that there is an infinite buffer space between the two machines. This assumption may not necessarily be realistic for some scheduling problems, e.g., see [36–38]. Therefore, another possible research area is to address the problem with a limited buffer space between the two machines.

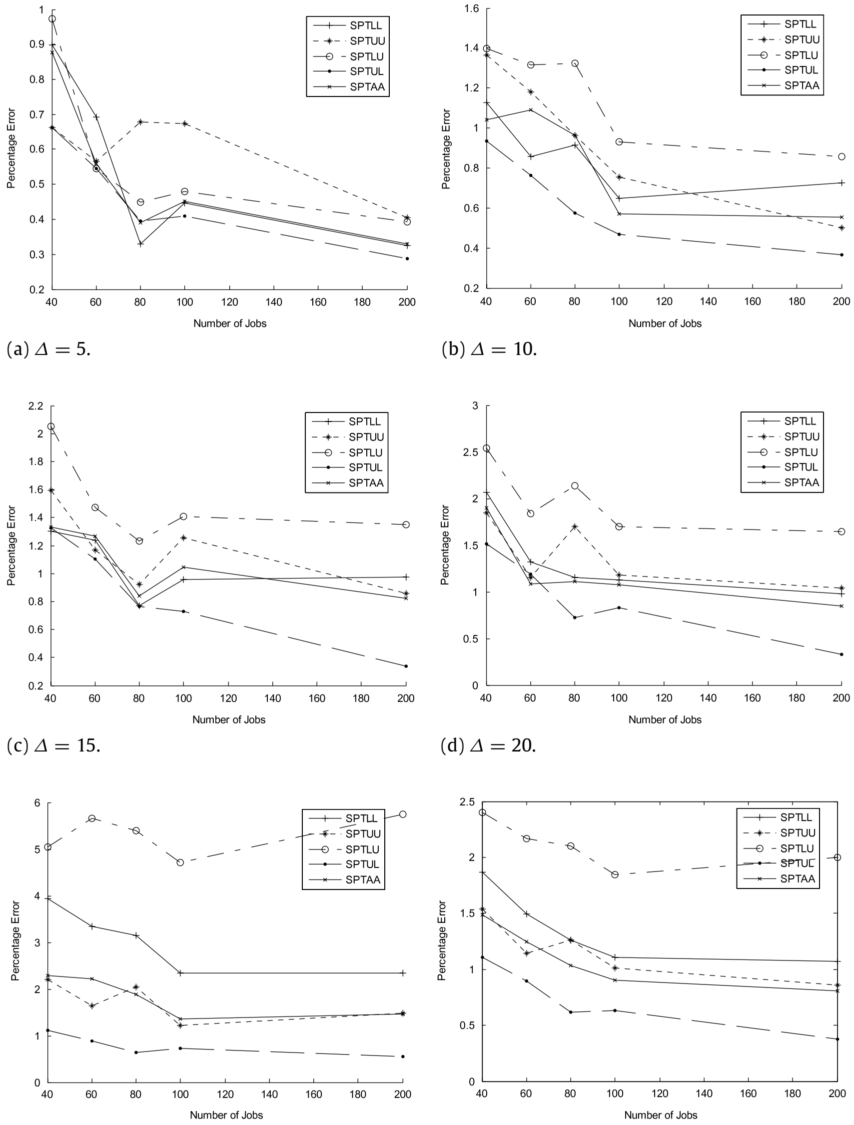
Scheduling problems with random and bounded processing times have been addressed in flowshop and jobshop environment but not in single or parallel machine environments, e.g., [39–41]. Therefore, single or parallel machine problems can be addressed with random and bounded processing times.

Acknowledgements

The authors would like to thank anonymous referees for their useful and constructive comments and suggestions.

Appendix

In this Appendix, we describe the distributions that have been used in Section 4 to evaluate the performances of the proposed heuristics. We considered three distributions; Uniform, Exponential (negative and positive), and Normal. The



(a) $\Delta = 5$. (b) $\Delta = 10$.

(c) $\Delta = 15$. (d) $\Delta = 20$.

(e) $\Delta = 40$. (f) Average over Δ.

Fig. 4. Average percentage error for positive exponential distribution.

considered uniform distribution is the same as regular uniform distribution. However, 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] O.P. Singh, S. Chand, Supply chain design in the electronics industry incorporating JIT purchasing, European Journal of Industrial Engineering 3 (2009) 21–44.
[2] R.D.H. Warburton, EOQ extensions exploiting the Lambert W function, European Journal of Industrial Engineering 3 (2009) 45–69.
[3] S. Sharma, On price increases and temporary price reductions with partial backordering, European Journal of Industrial Engineering 3 (2009) 70–89.
[4] 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.
[5] 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.

[6] 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.
[7] 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.
[8] T. Gonzalez, S. Sahni, Flow shop and job shop schedules, Operations Research 26 (1978) 36–52.
[9] H.M. Soroush, Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs, European Journal of Operational Research 113 (1999) 405–468.
[10] H.M. Soroush, Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem, European Journal of Operational Research 181 (2007) 266–287.
[11] A.A. Cunningham, S.K. Dutta, Scheduling jobs with exponentially distributed processing times on two machines of a flow shop, Naval Research Logistics Quarterly 16 (1973) 69–81.
[12] P.S. Ku, S.C. Niu, On Johnson's two-machine flow shop with random processing times, Operations Research 34 (1986) 130–136.
[13] P.J. Kalczynski, J. Kamburowski, A heuristic for minimizing the expected makespan in two-machine flowshops with consistent coefficients of variation, European Journal of Operational Research 169 (2006) 742–750.
[14] V. Portougal, D. Trietsch, Makespan-related criteria for comparing schedules in stochastic environments, Journal of the Operational Research Society 49 (1998) 1188–1195.
[15] V. Portougal, D. Trietsch, Stochastic scheduling with optimal customer service, Journal of the Operational Research Society 52 (2001) 226–233.
[16] V. Portougal, D. Trietsch, Johnson's problem with stochastic processing times and optimal service level, European Journal of Operational Research 169 (2006) 751–760.
[17] A. Allahverdi, J. Mittenthal, Scheduling on a two-machine flowshop subject to random breakdowns with a makespan objective function, European Journal of Operational Research 81 (1995) 376–387.
[18] G. Celano, A. Costa, S. Fichera, An evolutionary algorithm for pure fuzzy flowshop scheduling problems, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 11 (2003) 655–669.
[19] 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.
[20] 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.
[21] A. Barve, A. Kanda, R. Shankar, The role of human factors in agile supply chains, European Journal of Industrial Engineering 3 (2009) 2–20.
[22] 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.
[23] A. Hornung, L. Monch, Heuristic approaches for determining minimum cost delivery quantities in supply chains, European Journal of Industrial Engineering 2 (2008) 377–400.
[24] T.C Lai, Y.N. Sotskov, N.Y. Sotskova, F. Werner, Optimal makespan scheduling with given bounds of processing times, Mathematical and Computer Modeling 26 (1997) 67–86.
[25] T.C. Lai, Y.N. Sotskov, Sequencing with uncertain numerical data for makespan minimization, Journal of the Operational Research Society 50 (1999) 230–243.
[26] 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.
[27] 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 Modeling 49 (2009) 991–1011.
[28] R.L. Bregman, Dynamically allocating expediting funds in projects with schedule uncertainty, European Journal of Industrial Engineering 3 (2009) 363–376.
[29] Y. Sotskov, A. Allahverdi, T.C. Lai, Flowshop scheduling problem to minimize total completion time with random and bounded processing times, Journal of the Operational Research Society 55 (2004) 277–286.
[30] A. Allahverdi, Two-machine flowshop scheduling problem to minimize total completion time with bounded setup and processing times, International Journal of Production Economics 103 (2006) 386–400.
[31] A. Allahverdi, M.F. Tatari, Stochastic machine dominance in flowshops, Computers and Industrial Engineering 32 (1997) 735–741.
[32] Q.-K. Pan, L. Wang, A novel differential evolution algorithm for no-idle permutation flow-shop scheduling problem, European Journal of Industrial Engineering 2 (2008) 279–297.
[33] A. Allahverdi, J.N.D. Gupta, T. Aldowaisan, A review of scheduling research involving setup considerations, OMEGA The International Journal of Management Science 27 (1999) 219–239.
[34] 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.
[35] 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.
[36] 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.
[37] B. Qian, L. Wang, D.X. Huang, X. Wang, An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers, International Journal of Production Research 47 (2009) 1–24.
[38] A. Chelbi, D. Ait-Kadi, M. Radhoui, An integrated production and maintenance model for one failure prone machine-finite capacity buffer system for perishable products with constant demand, International Journal of Production Research 46 (2008) 5427–5440.
[39] 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.
[40] 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.
[41] 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.