

MINIMIZING MAKESPAN IN FLOWSHOPS WITH BOUNDED PROCESSING TIMES

A. Allahverdi¹, H. Aydilek²

¹Department of Industrial and Management Systems Engineering, Kuwait University, Kuwait

²Department of Natural Sciences and Mathematics, Gulf University for Science and Technology, Kuwait
ali.allahverdi@ku.edu.kw, aydilek.h@guest.edu.kw

ABSTRACT : *The two-machine flowshop scheduling problem of minimizing makespan is addressed where jobs have random processing times which are bounded within certain intervals. The probability distributions of job processing times within intervals are not known. The only known information about job processing times is the lower and upper bounds. The decision about a solution of the problem, i.e., finding a sequence, has to be made based on these bounds. Different heuristics using the bounds are proposed, and the proposed heuristics are compared based on randomly generated data. The computational analysis has shown that three of the proposed heuristics perform well with an overall average error of less than one percent. Moreover, it has also shown that one of the heuristics, which applies Johnson's algorithm to the average of the lower and upper bounds, perform as the best with an overall average percentage error of 0.71. The obtained results are also shown to be consistent with the recent results obtained in the literature.*

KEYWORDS: *Scheduling, flowshop, makespan, random and bounded processing times*

1 INTRODUCTION

The deterministic two-machine flowshop scheduling problem to minimize makespan is one of the most studied scheduling problems. Johnson (1954) established a polynomial time algorithm that finds an optimal solution to this problem. It is perfectly valid to assume that job processing times are deterministic for some scheduling environments for which Johnson (1954) algorithm can be used to solve the problem. However, for some other scheduling environments, the assumption of deterministic processing times is not valid. As pointed out by Soroush (1999, 2007), the random variation in processing times, has to be taken into account while searching for a solution. Some researchers addressed the flowshop problem where job processing times follow certain probability distributions. For example, Cunningham and Dutta (1973) and Ku and Niu (1986) addressed the problem where jobs have exponentially distributed processing times while Kalczynski and Kamburowski (2006) addressed the problem for the case where job processing times follow Weibull distribution.

Portougal and Trietsch (1998) suggested that variance reduction should be also taken into account while selecting a solution for stochastic flowshops. Portougal and Trietsch (2001) 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 (2006) developed and evaluated two heuristics. Their computational analysis indicated that one of their heuristics, which applies Johnson's algorithm to average processing times, is almost surely asymptotically optimal by expectation.

For some scheduling environments, even when job processing times are deterministic, job completion times may be stochastic as a result of random machine breakdowns. Allahverdi (1995, 1997) and Allahverdi and

Mittenthal (1994, 1995) addressed the two-machine flowshop scheduling problem where machines are subject to random breakdowns. Moreover, some researchers have recently proposed the use of a fuzzy set theory to model the uncertainty in scheduling problems, e.g., Celano et al. (2003).

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. (1997), and later studied by Lai and Sotskov (1999).

For the two-machine flowshop scheduling problem with bounded processing times to minimize makespan, Allahverdi and Sotskov (2003) provided some dominance relations. These dominance relations help reduce 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. Matsveichuk et al. (2008) 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. However, 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 lower and upper bounds being close to each other..

2 HEURISTICS

Let $L_{t_{j,m}}$ and $U_{t_{j,m}}$ 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, $L_{t_{j,m}} \leq t_{j,m} \leq U_{t_{j,m}}$. If $L_{t_{j,m}} = t_{j,m} = U_{t_{j,m}}$ for all jobs and both machines, then it is known that Johnson's algorithm minimizes makespan. 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 are processed on both machines.

When $U_{t_{i,j}} = L_{t_{i,j}}$ for all $i=1, 2, \dots, n$ and $j=1, 2$, the problem reduces to the deterministic two-machine flowshop scheduling problem for which an optimal solution can be obtained by Johnson's algorithm. In general, $U_{t_{i,j}} \neq L_{t_{i,j}}$ for at least some of the jobs. Moreover, 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., $U_{t_{i,j}}$ and $L_{t_{i,j}}$. 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 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 simple heuristics which utilize $U_{t_{i,j}}$ and $L_{t_{i,j}}$.

Let $a_{i,j} = (1-\lambda)L_{t_{i,j}} + \lambda U_{t_{i,j}}$ where $0 \leq \lambda \leq 1$. Notice that since the only available information about a job processing time is obtained from the lower and upper bounds, a decision about the position of a job in a sequence can be made by using $a_{i,j}$. The variable λ can be considered as the weight given to the lower and upper bounds. We consider five different values for λ , namely, 0, 0.25, 0.5, 0.75, 1. Once the value of λ is selected, then $a_{i,j}$ are computed. Now we have a single variable, i.e., $a_{i,j}$, for each job. Then, we use these $a_{i,j}$ as processing times and use Johnson's algorithm to obtain a sequence. Since there are five different values for λ , we will have five different heuristics which we call Heuristic 1 (H1) when $\lambda=0$, Heuristic 2 (H2) when $\lambda=0.25$, Heuristic 3 (H3) when $\lambda=0.5$, Heuristic 4 (H4) when $\lambda=0.75$, and Heuristic 5 (H5) when $\lambda=1$. It should

be noted that H1 uses the information only on the lower bounds while H5 uses the information only on the upper bounds. Similarly, H3 uses average values of the lower and upper bounds.

3 COMPUTATIONAL EXPERIMENTS

The proposed heuristics H1, H2, H3, H4, and H5 are evaluated based on randomly generated data following different distributions. To measure the effectiveness of the heuristics, we compared the performance of the proposed heuristics with that of Johnson's solution applied to the realized processing times. In other words, once the processing times are known (realized), then Johnson's algorithm can be applied to these realized processing times to obtain the optimal solution. We compared the performance of the heuristics using two measures: average percentage error (Error) and standard deviation (Std) out of one thousand replicates. The percentage error is defined as $100 * (\text{Makespan of the heuristic} - \text{Makespan of Johnson}) / (\text{Makespan of Johnson})$.

The lower and upper bounds of processing times are generated from uniform distributions such that $L_{t_{i,j}} \in U(1, \Delta)$ and $U_{t_{i,j}} \in U(\Delta, 2\Delta)$ where $\Delta \in \{10, 20, 40, 50, 100\}$. 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), normal (five sigma and seven sigma), and linear (negative and positive). For the normal distribution the mean is computed as $(U_{t_{i,j}} + L_{t_{i,j}}) / 2$ and the standard deviations are calculated as $(U_{t_{i,j}} - L_{t_{i,j}}) / 5$ and $(U_{t_{i,j}} - L_{t_{i,j}}) / 7$ for Normal (5σ) and Normal (7σ), respectively. 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 computational results of the evaluation of the heuristics H1-H5 are summarized in Figures 1-9. For the sake of brevity, only the summary results are given in the Figures. It should be noted that the standard deviations (Std), out of one thousand replicates, were significantly small. Moreover, comparison 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 Figure 1 it can be seen that H1, H2, and H3 perform better than H4 and H5. Moreover, H4 performs better than H5. This is expected for the negative exponential distribution as it is more likely that the

realization of processing time would be closer to the lower bound. This is also consistent with the theoretical results. It can also be seen that as the number of jobs increases the difference between errors of H1-H5 converges when Δ is relatively small. On the other hand, for positive exponential distribution (Figure 2), H1 performs as the worst heuristic followed by H2. The performances of the other three heuristics of H3-H5 are close to each other but H4 performs the best followed by H5. Again the results are consistent with the theoretical results.

Figures 3 and 4 indicate the results for negative and positive linear distributions, respectively. H2 performs as the best for the negative linear distribution while H5 performs as the worst heuristic. On the other hand, H4 is the best heuristic and H1 is the worst heuristic for the positive linear distribution. The results, in general, are consistent with theoretical results.

The results for uniform distribution are summarized in Figure 5. The figure shows that H2-H4 have performance close to each other while H1 and H5 perform worse than H2-H4. From H2-H4, H3 in general perform the best. Again the results are consistent with those theoretical results.

Figures 6 and 7 summarize the results for normal distribution with 7σ and 5σ , respectively. Both figures show similar results. Similar to previous results, H1 and H5 perform worst while the performances of H2-H4 are close to each other. Among H2-H4, H3 performs better than the others in general. This is expected, since it is more likely that a value between the lower and the upper bounds will be realized. The results are also consistent with the theoretical results.

The average results over all the considered distributions (negative and positive exponential, negative and positive linear, uniform, normal with 5σ and 7σ) are summarized in Figure 8 with respect to the number of jobs.

The performances of the heuristics are compared for different values of Δ in Figure 9. As can be seen from the figure, the performances of H1 and H5 are sensitive to the distribution, and hence, they are affected by Δ range. For example, for negative exponential distribution, the performance of H5 gets worst as Δ increases. For some other cases, the performance of H1 gets worst. It can be stated that the performances of H2-H4 are less sensitive to Δ while those of H1 and H5 are sensitive to Δ depending on the distribution.

In summary, H1 and H5 perform as the worst among the five heuristics with overall average errors of 1.381 and 1.372 percent, respectively. As can be seen from the figure, the overall error is less than one percent for H2, H3, and H4. More precisely, the overall average percentage errors of H2, H3, and H4 are 0.846, 0.707, and 0.803, respectively. This indicates that the heuristics H2-H4 are very efficient. Among the three efficient heuristics H2-H4, H3 is the best. As a result, it can be stated that Heuristic H3 is the best performing heuristic among all

the five heuristics. H3 is obtained by using Johnson algorithm based on the average of the lower and upper bounds as processing times. This shows that Johnson algorithm, based on the average of lower and upper bounds, can be successfully applied to the scheduling environments with random and bounded processing times. This result is also consistent with the conclusion of Portugal and Trietsch (2006) who considered random processing times without lower and upper bounds.

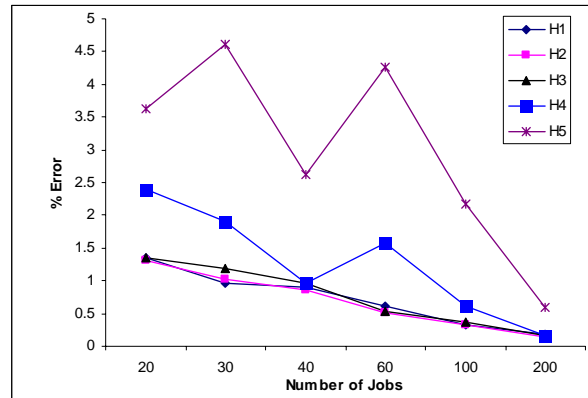


Fig. 1. Heuristic Comparisons for Negative Exponential Distribution for average Δ .

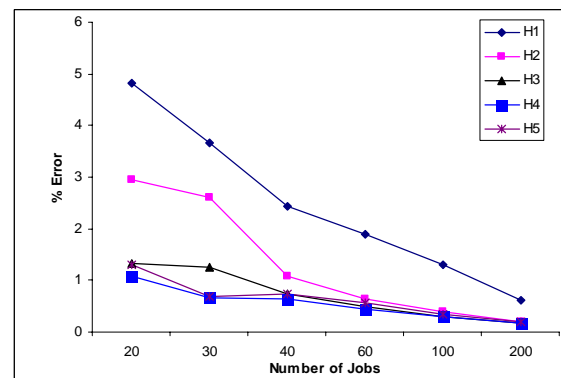


Fig. 2. Heuristic Comparisons for Positive Exponential Distribution for average Δ .

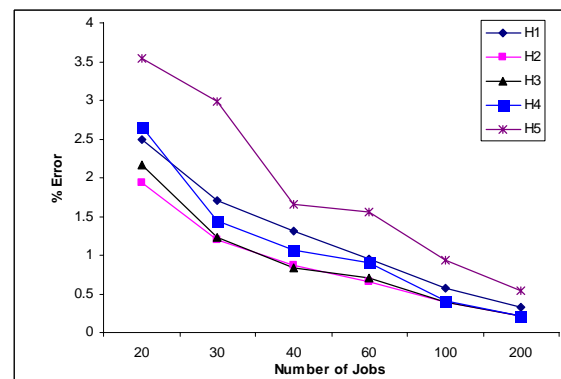


Fig. 3. Heuristic Comparisons for Negative Linear Distribution for average Δ .

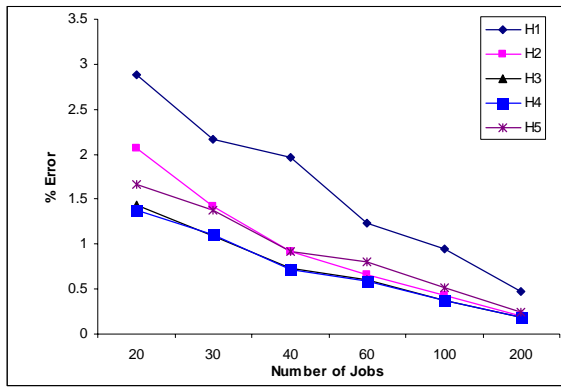


Fig. 4. Heuristic Comparisons for Positive Linear Distribution for average Δ .

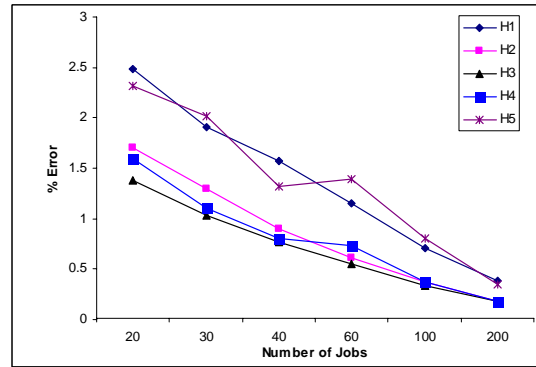


Fig. 8. Heuristic Comparisons for Average of Distributions

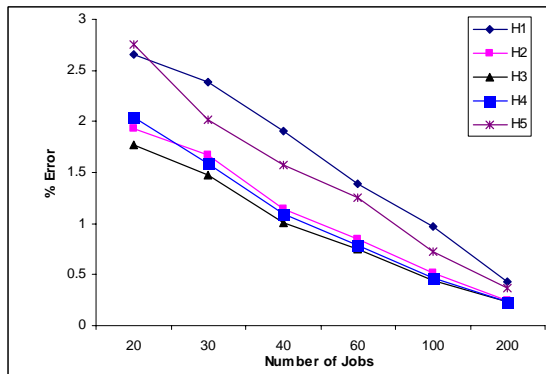


Fig. 5. Heuristic Comparisons for Uniform Distribution for average Δ .

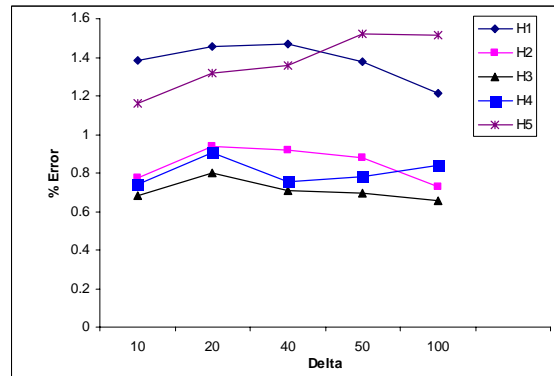


Fig. 9. Overall average error versus Δ

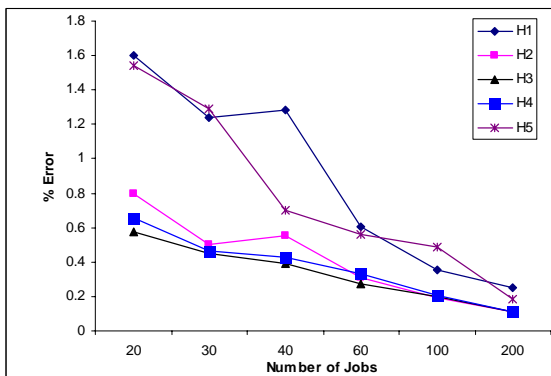


Fig. 6. Heuristic Comparisons for Normal Distribution (7σ) for average Δ .

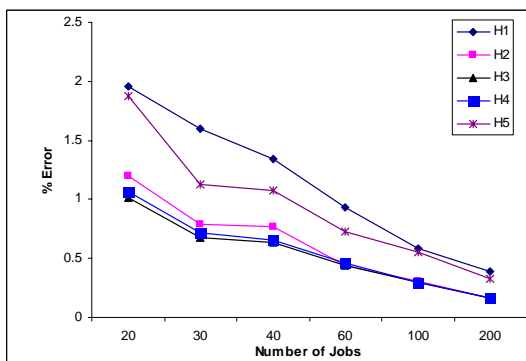


Fig. 7. Heuristic Comparisons for Normal Distribution (5σ) for average Δ .

4 CONCLUSION

The problem of scheduling on a two machine flowshop to minimize makespan was addressed in this paper. 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. Five different heuristics were proposed by taking into consideration the lower and upper bounds of job processing times. The performance of the heuristics was evaluated through an extensive computational experimentation. One of the heuristics is to use Johnson's algorithm which is based on the average of lower and upper bounds of job processing times, which may be called modified Johnson's algorithm. The computational experiments indicate that this heuristic performs as the best among the considered heuristics. Hence, the modified Johnson's algorithm can be successfully applied to the scheduling environments with random and bounded processing times.

There are many possible extension to the problem addressed in this paper. One possible extension is to address the problem with respect to mean completion time criterion or maximum lateness criterion. Another possible extension would be to consider no-idle flowshops, e.g., Pan and Wang (2008).

The importance of setup times has been addressed by

Allahverdi et al. (1999, 2008). 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 Vinod and Sridharan (2009). Therefore, another possible extension is to consider the problem addressed in this paper with setup times.

REFERENCES

- [1] Allahverdi, A., 1995. Two-stage production scheduling with separated setup times and stochastic breakdowns. *Journal of the Operational Research Society* 46, 896 - 904.
- [2] Allahverdi, A., 1997. Scheduling in stochastic flowshops with independent setup, processing, and removal times. *Computers & Operations Research* 24, 955 - 960.
- [3] Allahverdi, A., Gupta, J.N.D., Aldowaisan, T., 1999. A review of scheduling research involving setup considerations. *OMEGA The International Journal of Management Sciences* 27, 219-239.
- [4] Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY., 2008. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* 187, 985-1032.
- [5] Allahverdi, A., Mittenthal, J., 1994. Two-machine ordered flowshop scheduling under random breakdowns. *Mathematical and Computer Modeling* 20, 9 - 17.
- [6] Allahverdi, A., Mittenthal, J., 1995. Scheduling on a two-machine flowshop subject to random breakdowns with a makespan objective function. *European Journal of Operational Research* 81, 376 - 387.
- [7] Allahverdi, A., and Sotskov, Y.N., 2003. Two-machine flowshop minimum length scheduling problem with random and bounded processing times. *International Transactions in Operational Research* 10, 65-76.
- [8] Celano, G., Costa, A., Fichera, S. 2003. An evolutionary algorithm for pure fuzzy flowshop scheduling problems. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems* 11, 655-669.
- [9] Cunningham, A.A., Dutta, S.K., 1973. Scheduling jobs with exponentially distributed processing times on two machines of a flow shop. *Naval Research Logistics Quarterly* 16, 69-81.
- [10] Johnson S.M., 1954. Optimal two-and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* 1, 61 - 68.
- [11] Kalczyński, P.J., Kamburowski, J. 2006. A heuristic for minimizing the expected makespan in two-machine flowshops with consistent coefficients of variation. *European Journal of Operational Research* 169, 742-750.
- [12] Ku, P.S., Niu, S.C., 1986. On Johnson's two-machine flow shop with random processing times. *Operations Research* 34, 130 - 136.
- [13] Lai, T.C., Sotskov, Y.N., 1999. Sequencing with uncertain numerical data for makespan minimization. *Journal of the Operational Research Society* 50, 230 - 243.
- [14] Lai T.C, Sotskov, Y.N., Sotskova, N.Y., Werner, F., 1997. Optimal makespan scheduling with given bounds of processing times. *Mathematical and Computer Modeling* 26, 67 - 86.
- [15] Matsveichuk, N.M., Sotskov, Y.N., Egorova, N.G., and Lai, T.C., 2008. Schedule execution for two-machine flowshop with interval processing times. *Mathematical and Computer Modeling*, In Press.
- [16] Pan, Q-K, Wang, L., 2008. A novel differential evolution algorithm for no-idle permutation flow-shop scheduling problem. *European Journal of Industrial Engineering* 2, 279-297.
- [17] Portougal, V., and Trietsch, D. 1998. Makespan-related criteria for comparing schedules in stochastic environments. *Journal of the Operational Research Society* 49, 1188-1195.
- [18] Portougal, V., and Trietsch, D. 2001. Stochastic scheduling with optimal customer service. *Journal of the Operational Research Society* 52, 226-233.
- [19] Portougal, V., and Trietsch, D. 2006. Johnson's problem with stochastic processing times and optimal service level. *European Journal of Operational Research* 169, 751-760.
- [20] Soroush, H. M., 1999. Sequencing and due-date determination in the stochastic single machine problem with earliness and tardiness costs. *European Journal of Operational Research* 113, 405-468.
- [21] Soroush, H. M., 2007. Minimizing the weighted number of early and tardy jobs in a stochastic single machine scheduling problem. *European Journal of Operational Research* 181, 266-287.
- [22] Vinod, V., Sridharan, R., 2009. Simulation-based metamodels for scheduling a dynamic job shop with sequence-dependent setup times. *International Journal of Production Research* 47, 1425-1447.