8th International Conference of Modeling and Simulation - MOSIM'10 - May 10-12, 2010 - Hammamet - Tunisia
"Evaluation and optimization of innovative production systems of goods and services"

MINIMIZING MEAN COMPLETION TIME IN FLOWSHOPS WITH RANDOM PROCESSING TIMES

H. Aydilek[1], A. Allahverdi[2],
[1] Department of Natural Sciences and Mathematics, Gulf University for Science and Technology, Kuwait
[2] Department of Industrial and Management Systems Engineering, Kuwait University, Kuwait
aydilek.h@guest.edu.kw, ali.allahverdi@ku.edu.kw

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.*

KEYWORDS : *Scheduling, flowshop, total completion time, random and bounded processing times*

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 deterministic two-machine flowshop scheduling problem is unary NP-hard, e.g. see Gonzalez and Sahni (1978). 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 (1999, 2007), 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 (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.

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 different researchers including Lai and Sotskov (1999), Allahverdi and Sotskov (2003), and Matsveichuk et al. (2008). 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.

For the two-machine flowshop scheduling problem with bounded processing times to minimize total completion time, Sotskov et al. (2004) and Allahverdi (2006) 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.

2 HEURISTICS

Let Ltj,m and Utj,m denote the lower bound and the upper bound of processing time of job j on machine m, respectively. The exact value tj,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.

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 Uti,j=Lti,j for all i=1, 2, …, n and j=1, 2, the problem reduces to the deterministic two-machine flowshop scheduling problem. 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., Gonzalez and Sahni (1978). Moreover, Uti,j≠ Lti,j for at least some of the jobs, and the exact realization of ti,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 ti,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., Uti,j and Lti,j. It should be also noted that it does not make sense to use meta-heuristics since the exact values of ti,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 Uti,j and Lti,j.

We can use the idea of SPT while searching for heuristics. However, the exact values of ti,j are not known while the lower and upper bounds of ti,j, i.e., Uti,j and Lti,j are known. Hence, we can use the idea of SPT by using Uti,j and Lti,j values. For example, a sequence can be obtained by ordering the jobs according to SPT based on Lti,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 Uti2. This sequence is called SPTU2. By using Uti,j and Lti,j values, nine other sequences can be obtained. Table 1 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 Lti,1 and Uti,2. The rest of the heuristics are described in Table 1.

Table 1. Description of the proposed eleven heuristics

Heuristic Name	[Order jobs based on SPT according to]
SPTL1	[Lti,1]
SPTU1	[Uti,1]
SPTA1	[(Lti,1+ Uti,1)/2]
SPTL2	[Lti,2]
SPTU2	[Uti,2]
SPTA2	[(Lti,2+ Uti,2)/2]
SPTLL	[(Lti,1+ Lti,2)/2]
SPTUU	[(Uti,1+ Uti,2)/2]
SPTLU	[(Lti,1+ Uti,2)/2]
SPTUL	[(Uti,1+ Lti,2)/2]
SPTAA	[[(Lti,1+ Uti,1)/2+(Lti,2+ Uti,2)/2]/2]

3 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 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 Δ (*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.

The computational results indicated that for uniform distribution 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 for uniform distribution are summarized in Figure 1 for the well performing heuristics. Moreover, for the sake of brevity, the summary results are omitted for other distributions, and only the results will be discussed. It should be noted that the standard deviations (Std), out of two 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.

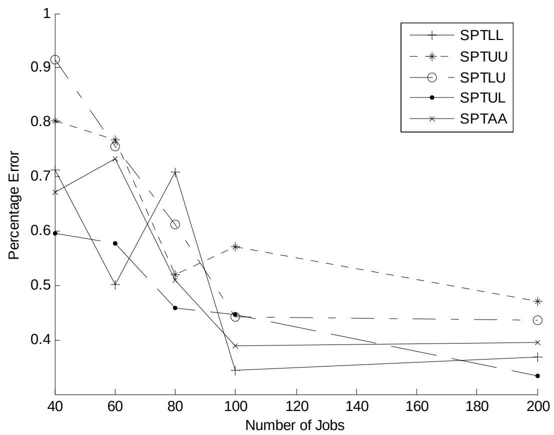
The results have revealed that the heuristics *SPTUL* and *SPTAA*, in general, perform better than the other three heuristics of *SPTLL*, *SPTUU*, and *SPTLU*. The good performance of *SPTAA* is not surprising since the sequence of jobs is determined based on the average of both lower and upper bounds 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 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.

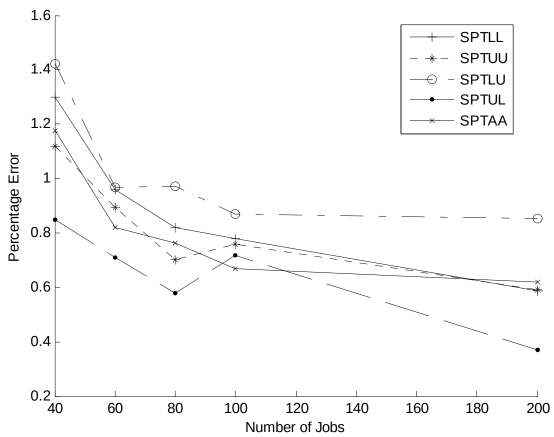
The difference between the performance of the heuristics of *SPTLL*, *SPTUU*, *SPTLU*, *SPTUL*, *SPTAA* gets larger as *Δ* gets larger. This is expected since as *Δ* approaches to zero, then the lower and upper bounds of job processing times approach to one another in which case all heuristics will yield 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.

The results have indicated 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.

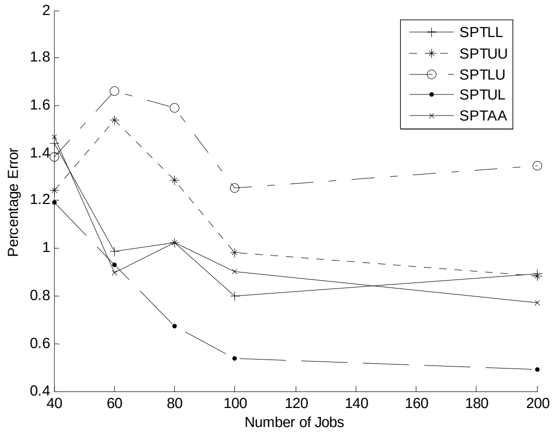
In summary, the heuristics taking into account the lower and upper bounds of job processing times on both machines perform much better than those which take into account the lower and upper bounds on one machine only. Furthermore, among those taking into account both bounds on both machines, SPTUL performs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

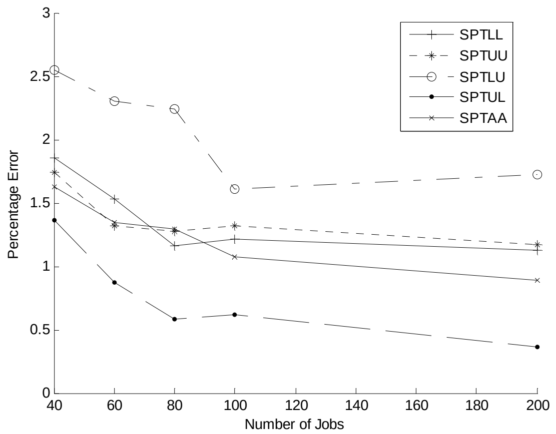
a) Δ = 5 and Δ = 10



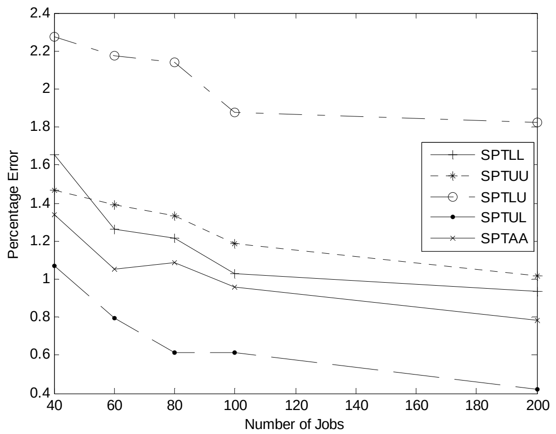


b) Δ = 15 and Δ = 20





c) Δ = 40 and Average over Δ



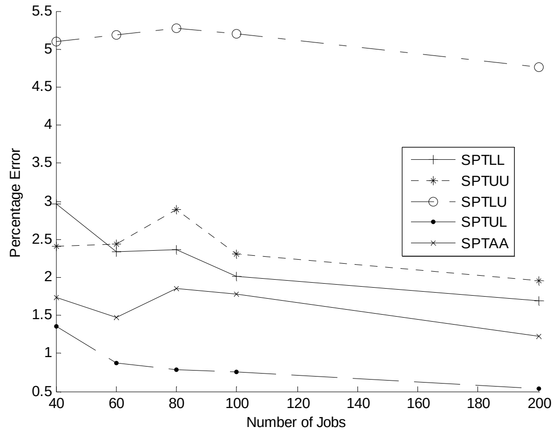


Figure 1. Average Percentage Error for Uniform Distribution

4 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 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.

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. Therefore, another possible extension is to consider the problem addressed in this paper with setup times.

REFERENCES

[1] Allahverdi, A., 2006. Two-machine flowshop scheduling problem to minimize total completion time with bounded

setup and processing times. International Journal of Production Economics 103, 386-400.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] Gonzalez, T., Sahni, S., 1978. Flow shop and job shop schedules. Operations Research 26, 36-52.

[7] Kalczynski, 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.

[8] Ku, P.S., Niu, S.C., 1986. On Johnson's two-machine flow shop with random processing times. Operations Research 34, 130 - 136.

[9] Lai, T.C., Sotskov, Y.N., 1999. Sequencing with uncertain numerical data for makespan minimization. Journal of the Operational Research Society 50, 230 - 243.

[10] 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.

[11] Matsveichuk, N.M., Sotskov, Y.N., Egorova, N.G., and Lai, T.C., 2008. Schedule execution for two-machine flow-shop with interval processing times. Mathematical and Computer Modeling, In Press.

[12] Portougal, V., and Trietsch, D. 1998. Makespan-related criteria for comparing schedules in stochastic environments. Journal of the Operational Research Society 49, 1188-1195.

[13] Portougal, V., and Trietsch, D. 2001. Stochastic scheduling with optimal customer service. Journal of the Operational Research Society 52, 226-233.

[14] 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.

[15] Qian, B., Wang, L., Huang, D. X., Wang, X., 2009. An effective hybrid DE-based algorithm for flow shop scheduling with limited buffers. International Journal of Production Research 47, 1-24.

[16] 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.

[17] 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.

[18] Sotskov, Y., Allahverdi, A., Lai, T.C., 2004. Flowshop scheduling problem to minimize total completion time with random and bounded processing times. Journal of the Operational Research Society 55, 277-286.