HEURISTIC SOLUTION METHODS FOR THE STOCHASTIC FLOW SHOP PROBLEM

By

Kenneth R. Baker

Tuck School of Business
Dartmouth College
Hanover, NH, USA
Email: ken.baker@dartmouth.edu
Phone: 603-646-2064

and

Dominik Altheimer

Helmut Schmidt Universitat
Hamburg, GERMANY

December, 2010
Revised July, 2011

Abstract

We investigate the stochastic flow shop problem with *m* machines and general distributions for processing times. No analytic method exists for solving this problem, so we looked instead at heuristic methods. We devised three constructive procedures with modest computational requirements, each based on approaches that have been successful at solving the deterministic counterpart. We compared the performance of these procedures experimentally on a set of test problems and found that all of them achieve near-optimal performance.

Keywords: stochastic scheduling, flow shop, makespan, lognormal distribution

HEURISTIC SOLUTION METHODS FOR THE STOCHASTIC FLOW SHOP PROBLEM

The flow shop problem plays an important role in the theory of scheduling. The deterministic version was introduced to the literature by Johnson (1954), in what is often identified as the first formal study of a problem in scheduling theory. That article has led to a large number of papers studying variations of the basic model and various algorithmic approaches for finding solutions. For example, Reisman et al. (1997) claimed to have located 170 articles containing contributions to the "subdiscipline" of flow shop scheduling. More recently, Ruiz and Maroto (2005) cited 53 articles in their review paper on heuristic procedures for the permutation flow shop problem with makespan objective. Framinan *et al*. (2004) cited 76 articles in a review paper on the same topic. Reza Hejazi and Saghafian (2005) cited 176 articles in a review paper on exact and heuristic approaches to the same problem. Clearly, the flow shop scheduling problem has attracted a lot of attention.

On the other hand, progress with the stochastic version of the flow shop problem has been very limited. Few general results have been obtained, and the optimization of basic cases remains a challenge. In this paper, we present a comparative study of heuristic methods for solving the *m*-machine stochastic flow shop problem with the objective of minimizing the expected makespan. We focus on a few relatively simple heuristic approaches that are motivated by the existing literature, and we compare their performance on a set of test problems. Finally, we summarize our results and suggest what questions might guide future research on this subject.

Background on the deterministic model

The classical flow shop problem contains *n* jobs and *m* machines, as well as a set of standard assumptions (see, for example, Baker and Trietsch, 2009a). The objective is to minimize the length of the schedule or *makespan*.) In the case of two machines, we can construct an optimal job sequence by employing Johnson's Rule (Johnson, 1954), which leads to an efficient algorithm. In the case of three or more machines, the flow shop problem is *NP*-hard. Several effective heuristic procedures have been invented for solving problems with three or more machines. Relatively recent reviews of that literature have been compiled by Framinan *et al.* (2004) and by Ruiz and Maroto (2005). We mention two heuristics in particular, as they are adapted to the stochastic model in our work. The first is due to Campbell, Dudek, and Smith (1970), known as the CDS Heuristic. The second is due to Nawaz, Enscore, and Ham (1983), known as the NEH Heuristic. Both are *constructive* heuristics. This term means that the algorithms perform a predictable amount of computation and ultimately construct a complete schedule. In contrast, an *improvement* heuristic starts with a given sequence and searches for improvement, but the computational effort is unpredictable. Improvement heuristics are usually based on generic methods such as neighborhood search. Sophisticated forms of improvement heuristics include tabu search, simulated annealing and genetic algorithms.

The CDS algorithm uses Johnson's Rule in a heuristic fashion and creates several schedules from which a "best" schedule is chosen. The algorithm corresponds to a multistage use of Johnson's Rule applied to a two-machine pseudo-problem derived from the original. The NEH algorithm constructs a single sequence, starting with a list of the jobs. The first two jobs on the list are removed, the two possible permutation sequences of those jobs are constructed, and the better of the two is retained (with ties broken arbitrarily). The relative sequence position of the first two jobs is then fixed. At each succeeding step, a job is removed from the list and placed optimally into the partial sequence retained from the previous step. When the last job is removed from the list, a full sequence is chosen from among the possible insertions at the final step.

The CDS algorithm and the NEH algorithm are computationally efficient. The computational complexity is $O(mn\log n)$ for the CDS algorithm and $O(mn^2)$ for the NEH algorithm. Comparative studies by Park et al. (1984), Widmer and Hertz (1989), Taillard (1990), Ho and Chang (1991), Ponnambalam, et al., (2001), and Ruiz and Maroto (2005) have tended to reinforce the proposition that these are the two best and most robust constructive procedures available.

The Stochastic Model

The most common stochastic version of the flow shop problem assumes that the processing times are allowed to be random variables. In particular, we assume that the processing time of job j on machine k follows a probability distribution with mean μ_{kj} and standard deviation σ_{kj} (denoted σ when it applies across all jobs and machines). For convenience, we also assume that the processing times are drawn independently from distributions of a given family, such as the normal or the uniform. As a result, the makespan will also be random, and the objective is to minimize its expected value. This single change from the deterministic version of the problem is sufficient to make the problem quite difficult to solve. In fact, no analytic solution procedure exists for the stochastic version. Little attention has even been paid to finding heuristic procedures for the stochastic flow shop problem, although Portugal and Trietsch (2006) have shown that Johnson's Rule applied to mean values will produce asymptotically optimal expected makespan values in the stochastic case. Our paper essentially presents the first study comparing heuristic procedures for the m-machine stochastic flow shop problem with expected makespan criterion.

If we restrict attention to the two-machine stochastic flow shop problem, it is still the case that no general results are known, but if we restrict ourselves further to the case of exponential distributions, then we have one result, which states: the expected makespan is minimized by sequencing the jobs in nonincreasing order of $(1/\mu_{i1} - 1/\mu_{i2})$. This ordering is known as Talwar's Rule. It was conjectured to be optimal by Talwar (1967) and later proven optimal by Cunningham and Dutta (1973). Thus, sequencing jobs based on the differences in their mean processing rates provides the optimal two-machine solution for one special case.

With the solution to the two-machine case established, we might look next to the m-machine case with exponential distributions, but generalizations of Talwar's Rule have not been developed for three or more machines. One advantage of the exponential assumption is the possibility of analytic calculation of the expected makespan. (Lacking an optimal sequencing rule, we must have such a capability merely to compare one sequence with another.) However, a disadvantage of the exponential distribution is the fact that it has only one parameter: its mean cannot be different from its standard deviation. Pinedo (1982) suggests the following rule of thumb: “Schedule jobs with smaller expected processing times and larger variances in the processing times toward the beginning and the end of the sequence.” But for this rule to have meaning, we must deal with distributions that have distinct mean and variance parameters, unlike the exponential. Kalczynski and Kamburowski (2006) heuristically adapted Talwar's result for Weibull distributions, but did not attempt to generalize beyond two machines.

Baker and Trietsch (2010) tested three simple heuristic procedures for the two-machine stochastic model with general probability distributions. They compared Johnson’s Heuristic (Johnson’s Rule applied to the mean processing times), Talwar’s Heuristic (Talwar’s Rule applied to the mean processing times), and an Adjacent Pairwise Interchange Heuristic (which swapped adjacent jobs if their sequence, when considered separately, could be improved). Although none of the heuristic procedures dominated the others, Baker and Trietsch found that they all achieved very good performance, providing expected makespan values that, on average, were within 1% of the best value found. In our work, we demonstrate that this same good performance can be achieved in the *m*-machine case.

For the exponential case, Gourgand et al. (2003) show that the expected makespan calculation can be carried out for *m* machines analytically using a Markovian approach, but even that method encounters limitations due to problem size. (They proceed no further in making the calculation than medium-size problems of 20 jobs and 5 machines.) They conclude that we must ultimately rely on simulation techniques to evaluate the expected makespan. Thus, to make progress on the model with general probability distributions for processing times, we shall have to rely on (1) heuristic procedures to find good sequences and (2) simulation procedures to calculate expected makespan values.

Heuristic Procedures

We describe three main heuristic procedures for sequencing jobs in the stochastic flow shop with expected makespan objective. Two of these procedures follow the logic of the CDS algorithm. In other words, they create a series of two-machine pseudo-problems; then those pseudo-problems are solved by a two-machine algorithm (either Johnson's Rule or Talwar's Rule). The procedures are thus referred to as Johnson’s Heuristic and Talwar’s Heuristic.

Johnson’s Heuristic solves a two-machine stochastic flow shop problem by replacing the processing times with their mean values. Then, the resulting deterministic problem is solved by Johnson’s Rule to deliver a desired sequence for the jobs. This procedure is heuristic because it solves a deterministic counterpart of the stochastic problem. Talwar’s Heuristic solves a two-machine stochastic flow shop problem by applying Talwar’s Rule (sorting the jobs by nonincreasing differences of the mean processing rates). This procedure is heuristic because its optimality does not extend to general distributions.

Thus, the first two heuristic procedures might be called the CDS/Johnson Heuristic and the CDS/Talwar Heuristic. Our third procedure is the NEH algorithm, applied to the stochastic problem directly. That is, the procedure finds the best two-job sequence; then, keeping the two jobs in their better order, it finds the best insertion of the third job into the two-job sequence, then the best insertion of the fourth job into the best three-job sequence, and so on. The jobs are considered in the order of nonincreasing total mean processing time.

Each heuristic procedure requires the ability to compare two job sequences and choose the better one. In other words, we must be able to find the expected makespan for each of two sequences in a given stochastic flow shop problem and identify the smaller of the two. For this purpose, we use simulation. Gourgand *et al*. (2003) assessed the accuracy of simulation by making comparisons in cases for which their Markovian analysis is practicable. They tested different sample sizes on a standard dataset and found, for example, that sample sizes of 200,000 produced 95% confidence intervals on the order of 0.1% and average estimation errors on the order of 0.05%. Those results were based on comparing simulation and analytic calculations for stochastic flow shop problems with exponential processing times. The tests used lexicographic job sequences (i.e., the equivalent of an arbitrary sequence) and indicated that a sample size this large is more than sufficient to obtain useful estimates.

Beyond the question of precision in simulation, one might ask whether the use of simulation with a heuristic method is likely to generate the same solution (that is, the same job sequence) as the implementation with Markovian analysis. Gourgand et al. explored this question with the CDS/Talwar Heuristic. Sometimes differences occurred, but on those occasions the deviations between the expected makespan values generated by the two sequences (and evaluated analytically) were less than 0.01% when they occurred. The authors also found that, for the purposes of generating the best sequence using the heuristic procedure, sample sizes of 5,000 were adequate. Larger sample sizes may lead to different *estimates*, but they seldom led to different *sequences*. Therefore, Gourgand et al. concluded that sample sizes of 5,000 were sufficient for testing heuristic procedures.

The work of Gourgand et al. suggests that two possible drawbacks to using simulation are not as severe as we might initially believe. First, simulation might lead us to the “wrong” sequence when we implement a heuristic procedure. Gourgand et al. observed that this outcome was unlikely. Second, simulation might lead us to an incorrect estimate of the expected makespan, even when we find the “right” sequence. Gourgand et al. observed that estimation error is quite small. Using their result for an arbitrary sequence, we estimate that the estimation error with simulation sample sizes of 100,000 is less than 0.01%. But their data comes from the exponential case, which carries larger estimation errors than we would expect in cases for which the standard deviation is less than the mean. Furthermore, we would expect smaller estimation errors in sequences produced by a heuristic procedure than for an arbitrary sequence because the variance associated with a stochastic makespan tends to be smaller for “good” sequences than for “poor” sequences, as discussed by Baker and Trietsch (2009a).

In our calculations, we used estimates based on simulation sample sizes of 100,000 (generated using the Latin Hypercube method to reduce variance), and we examined the results obtained by implementing heuristic procedures. Thus, the effects of the two types of simulation errors would appear to be relatively small.

Computational Experiments

We devised a set of experiments to compare the performance of the heuristic procedures introduced in the previous section. We created a set of stochastic flow shop problems in which the objective is to find the minimum expected makespan. To specify an instance of the problem, we had to specify the number of jobs and the processing time distributions for each job on each machine. For simplicity, we assumed that all processing time distributions are members of the same family, such as exponential, uniform, or lognormal. The exponential distribution is specified by a single parameter (its mean value) and corresponds to the case in which Talwar's Heuristic produces an optimal schedule for two machines. The uniform distribution is specified by two parameters (e.g., a mean value and a range) and allows us to influence the extent to which processing time distributions overlap. Finally, the lognormal distribution is the basis for most of the experimental work, for two reasons. First, the lognormal can also be specified by two parameters (a mean value and a standard deviation). Second, the lognormal has considerable practical value as a model for actual processing times. Among other traits, a lognormal variate is nonnegative, and its parameters can represent both high and low coefficients of variation.

The first step in generating a test problem was to obtain the parameters (the mean and, as needed, the standard deviation) for each of the processing time distributions in the problem. These parameters were drawn randomly as samples from a pre-specified interval. For example, we might obtain the mean processing times for six jobs by sampling $6m$ times from the interval between 10 and 20 and the standard deviations by sampling $6m$ times from the interval between 5 and 10.

Once the parameters of the processing time distributions were determined, the next step was to construct the job sequence using each of the heuristic methods. For comparison purposes, we also evaluated the sequence of jobs in numerical order. We can think of this sequence as "random" in that it uses no information about the processing time distributions in ordering the jobs.

The next step was to estimate the mean value of the makespan for each of the four sequences. This estimate was based on a simulation containing 100,000 trials. In addition, we used a fifth algorithm to search for an optimal sequence: the Evolutionary Solver.[1] The Evolutionary Solver is an advanced genetic algorithm, which we initialized with the best sequence found by any of the heuristic methods. We then executed one run of the Evolutionary Solver and took its result as the optimal solution, as a basis for calculating the suboptimality of the other sequences. Gourgand et al. found that improvement methods achieved better algorithmic performance than constructive heuristics, with average suboptimalities close to 0.1-0.2% as compared to about 1.0-1.5% for a CDS-type heuristic. Although there is no guarantee that the Evolutionary Solver produces an optimal solution in every instance, we expect that it comes close enough for the purposes of evaluating the three main heuristic procedures in our study.

This procedure was replicated 10 times, each time sampling for the parameters of the processing time distributions. We recorded the average deviation between the heuristic procedure’s estimated mean and the best value found (referred to as the "average suboptimality"), as well as the number of

---

[1] The Evolutionary Solver is a proprietary genetic algorithm which appears to be particularly effective at solving sequencing problems. It is available as an Excel add-in, part of Frontline Systems' software package Risk Solver Platform (see www.solver.com).

times (out of 10) that each heuristic produced the best value. This design parallels the design used in the study of the two machine problem (Baker and Trietsch, 2010).

Tables 1-5 present summaries of our experiments. Here, we describe the notation in our tables.

- The top row of the table shows the problem size and the family of distributions from which processing times were sampled.
- The second row, labeled *Means*, shows the interval of values from which the various mean values were sampled.
- The third row, labeled *Ranges* in the case of uniform distributions, shows the range of the individual processing time distributions. This row is omitted in the exponential case because the mean value describes the variability in the distribution. In the case of the lognormal distribution, this row is labeled σ for the standard deviation of the distribution or the interval of standard deviations sampled.
- The remaining rows are labeled according to the sequencing method: *Random, CDS/Johnson, CDS/Talwar,* and *NEH*. Across each row, the table contains four pairs of numbers, each pair giving our measure of average suboptimality and the number of times (out of 10) the procedure generated the best value.

Preliminary Experimental Results

To illustrate the type of data we compiled, we review some of our preliminary experiments, involving 10-job problems. (We tested a similar set of six-job problems and found results to be qualitatively the same.) We did not experiment with problems involving large numbers of jobs for two reasons. First, the computational requirements would be prohibitive. But perhaps more importantly, the asymptotic optimality results mentioned earlier suggest that problems with large numbers of jobs may not require stochastic sequencing: we can find a good sequence by addressing the deterministic counterpart.

In our first experiment, we used the uniform distribution as a model for processing times. We drew mean processing times as integers from the interval (40, 60) and took the range of the processing times to be 1. This is a special case in which we can anticipate the nature of the optimal solution for two machines. Except for the possibility that the means of two processing time distributions might match, the distributions do not overlap. In the two-machine case, where the CDS algorithm has only one stage, the CDS/Johnson Heuristic reduces to Johnson's Rule applied to the mean processing times. Furthermore, Johnson's Rule is known to be optimal in these cases because the stochastic problem is solved by its deterministic counterpart. Thus, we should expect the CDS/Johnson Heuristic to provide optimal solutions in every case with two machines and no overlap, which is what we observe in the validation experiments reported in one portion of Table 1. As the range of processing times is increased from 1 to as much as 10, 20, and 30, the performance of the CDS/Johnson Heuristic deteriorates slightly but with average suboptimalities averaging less than 0.1%. The CDS/Talwar Heuristic performs worse when no overlap occurs but improves as the range of processing times increases. The NEH Heuristic, which is virtually as good as the CDS/Johnson Heuristic in the nonoverlapping case, also deteriorates as the range increases.

The three-machine and six-machine results show some similarities, but the patterns that occurred in the two-machine instances do not appear consistently. In these instances, the NEH Heuristic performs best when no overlap exists, and the CDS/Talwar Heuristic improves as the range of processing times is increased. The overall picture suggests that each heuristic generates its share of optimal solutions but does not do so consistently.

The results in Table 1 also indicate that no dominance exists among the three heuristic procedures. None of the heuristic procedures generates optimal solutions for as many as half of the 120 test problems. However, the suboptimalities of the three heuristic procedures each lie below 1% on average, whereas the Random sequence generates average suboptimalities that are roughly 3-4% on average.

In the second set of preliminary experiments, we sampled from exponential distributions, pursuing another special case in which we can anticipate the optimal solution: In the two-machine case, the CDS/Talwar Heuristic reduces to Talwar's Rule, which is known to be optimal. We confirm this performance in the validation experiments of Table 2, where the CDS/Talwar Heuristic produces the best solution in all of the two-machine instances. With three machines, the CDS/Talwar Heuristic produces the best solution in 36 of the 40 instances, but with six machines, the CDS/Johnson Heuristic becomes more competitive. With exponential distributions, the Random sequence achieves average suboptimalities of 3-11%, generally worse than with uniform distributions, and the NEH Heuristic tends to be the worst of the three main heuristics being compared.

These preliminary results include some validation experiments, demonstrating that the few theoretical properties known for the stochastic flow shop problem are confirmed in our simulation results. However, neither the uniform distribution nor the exponential distribution reflects the probabilistic behavior we are likely to find in practical scheduling problems. For that reason, we turn to the lognormal distribution in our main experiments.

Main Experimental Results

In our main experiments, involving lognormal distributions, we were able to vary the means or the variances, or both, in the distributions of processing times. In the first set of experiments (Table 3), we sampled mean values from a fixed interval and varied the standard deviation from 1 to 5, 10, 20, 30, and 40. The results resemble those in Table 1. When $\sigma = 1$, the CDS/Johnson Heuristic is optimal in all of the two-machine instances, but performance deteriorates as σ increases. The NEH Heuristic produces very low average suboptimalities when $\sigma = 1$, for three-machine and six-machine cases as well, but its performance also deteriorates with increases in σ. The CDS/Talwar Heuristic provides some of the best average suboptimalities when σ is large, but it exhibits the worst average performance of the three heuristics when $\sigma = 1$. In addition, the Random sequence achieves average suboptimalities between 2% and 3%.

In Table 4, we summarize a complementary set of runs in which we fixed $\sigma = 10$ and varied the mean values. The results for two machines indicate that the CDS/Johnson Heuristic and the CDS/Talwar Heuristic achieve the best performance, with the latter slighly better when more overlap exists. The NEH Heuristic is consistently worse and achieves optimality only four times in the 20 test problems. For three and six machines, however, the NEH Heuristic is generally the best. Although the CDS/Johnson and CDS/Talwar Heuristics may be better when overlap is great, their average suboptimality becomes relatively large (above 1%) when overlap is reduced. In these runs, however, the average suboptimality of the Random sequence grows with increasing variability in the means, reaching a high of over 13% in the most variable case. Comparing the results in Table 4 to those in Table 3, we observe that performance is more sensitive to variations in the mean processing times than it is to the standard deviation of processing times.

Table 5 summarizes our most general experimental runs, in which both the means and the standard deviations were varied. In these 150 problem instances, each of the three heuristic procedures attained average suboptimalities of less than 1%, with the CDS/Talwar Heuristic lowest at 0.24%. (By comparison, the average for the Random sequence was about 6% in this data set.) Again, we find no overall dominance among the three main heuristic methods. The NEH Heuristic was best for the subset of instances with six machines. The CDS/Talwar Heuristic tended to be best for the 2 or three-machine problems, but this advantage disappeared in the six-machine problems.

Summary

We addressed the *m*-machine stochastic flow shop problem with expected makespan objective. At present, this problem cannot be solved by analytic techniques. The work of Gourgand et al. (2003) shows that even the special case of the exponential distribution encounters computational difficulties due to the combinatorial nature of the calculations. Therefore, it appears necessary to rely on simulation to estimate values of the objective function.

The limited study of exponential distributions carried out in Gourgand et al. (2003) also suggests that heuristic methods are able to produce schedules that often come within 1% of the optimal objective function. Therefore, if we can find a reliable heuristic procedure, we can expect it to generate near-optimal results. In the exponential case, their research indicated that general search methods could be even more effective.

We evaluated and compared three heuristic procedures. Each of the heuristic procedures is adapted from procedures that perform well in deterministic flow shop problems. Each of the heuristic procedures is a constructive procedure, meaning that it has modest (and predictable) computational requirements. We used a sophisticated improvement procedure—the evolutionary solver—to provide a proxy for the optimal solution, and we relied on simulation for evaluating schedules. Our main experiments consisted of 450 problem instances in which the processing times were drawn from lognormal distributions.

We know of only two special cases in which we could predict that one of the heuristic procedures would be optimal, and we used those cases to create validation experiments. When the probability distributions do not overlap, Johnson's Rule for the deterministic counterpart is optimal for two-machine problems. We confirmed this pattern in the case of uniform distributions with no overlap. However, this property does not generalize to three-machine and six-machine problems, for which we found that the CDS/Johnson Heuristic was not even the best average performer. A similar pattern tended to occur in the lognormal cases: the CDS/Johnson Heuristic was the best performer with very small σ in two-machine problems, but not necessarily in larger problems.

We also know that when all probability distributions are exponential, Talwar's Rule is optimal for two-machine problems. We confirmed this pattern as well; however, it does not necessarily generalize to three-machine and six-machine problems. In the lognormal cases, we might guess that the

CDS/Talwar Heuristic is desirable for large σ, but we found that the other heuristics were competitive, especially for six machines.

Overall, each of the heuristic procedures generated average suboptimalities less than 1%. For the 450 test problems in our main experiments, the CDS/Talwar Heuristic had an average suboptimality of 0.30%, with the CDS/Johnson Heuristic at 0.36% and the NEH Heuristic at 0.59%. (For the same problems, the Random rule, by comparison, had an average of 5.3%.) As the results in Tables 3-5 indicate, we found that none of the heuristic procedures dominated the others in all circumstances. Each was able to find best solutions for problem instances in which the other heuristics did not. Each of the heuristic procedures was able to find the best solution in about a third of the test problems.

Thus, a reliable analytic method for finding optimal solutions to the *m*-machine stochastic flow shop problem still eludes us. However, we can find solutions that are very close to optimal with the help of three constructive heuristics which lend themselves easily to practical implementation. Future research could investigate the benefits to be gained by using improvement methods and other sophisticated search algorithms, but since the three constructive heuristics already seem able to obtain solutions within 1% of optimality, little room exists for search algorithms to demonstrate improvement. A more fruitful direction might be to examine other objective functions, such as expected total flowtime or expected total tardiness. In yet another direction, future work could pursue notions of safe scheduling, as introduced by Baker and Trietsch (2009b), who suggest that schedules with small expected makespan values are likely to also perform well against tardiness objectives. In all of these directions, the stochastic flow shop model still represents a testing ground for new concepts in scheduling.

References

Baker, K.R. and D. Trietsch (2009a). *Principles of Sequencing and Scheduling*. Hoboken: Wiley.

Baker, K.R. and D. Trietsch (2009b). Safe scheduling: Setting due dates in single-machine problems. *European Journal of Operational Research* 196, 69-77.

Baker, K.R. and D. Trietsch (2010). Three Heuristic Procedures for the Stochastic, Two-Machine Flow Shop Problem. *Journal of* Scheduling. DOI 10.1007/s10951-010-0219-4

Campbell, H.G., R.A. Dudek, and M.L. Smith (1970). A heuristic algorithm for the *n* Job, *m* Machine sequencing problem. *Management Science,* 16, 630-637.

Cunningham, A.A. and S.K. Dutta (1973). Scheduling jobs with exponentially distributed processing times on two machines of a flow shop. *Naval Research Logistics Quarterly,* 16, 69-81**.**

Framinan, J.M., J.N.D. Gupta, and R. Leisten (2004). A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. *Journal of the Operational Research Society*, 55, 1243-1255.

Gourgand, M., N. Grangeon, and S. Norre (2003). A contribution to the stochastic flow shop scheduling problem. *European Journal of Operational Research* 151, 415-433.

Ho, J.C. and Y.-L. Chang (1991). A new heuristic for the *n*-job, *m*-machine flow-shop problem. *European Journal of Operational Research* 52, 194–202.

Johnson, S.M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1, 61-68.

Kalczynski, P.J. and J. Kamburowski (2006) A heuristic for minimizing the expected makespan in two-machine flow shops with consistent coefficients of variation, *European Journal of Operational Research* 169, 742-750.

Nawaz, M., E. Enscore Jr, and I. Ham (1983). A heuristic algorithm for the m-machine, n-job flowshop sequencing problem. *Omega* 11, 91–95.

Park, Y.B., C.D. Pegden, and E.E. Enscore (1984). A survey and evaluation of static flowshop scheduling heuristics. *International Journal of Production Research* 22, 127-141.

Pinedo, M. (1982). Minimizing the expected makespan in stochastic flow shops. *Operations Research* 30, 148-162.

Ponnambalam, S. G., P. Aravindan, and S. Chandrasekaran (2001). Constructive and improvement flow shop scheduling heuristics: an extensive evaluation. *Production Planning & Control* 12, 335-344.

Portougal, V. and D. Trietsch (2006). Johnson's problem with stochastic processing times and optimal service level. *European Journal of Operational Research,* 169, 751-760.

Reisman, A., A. Kumar, and J. Motwani (1997). Flowshop scheduling/sequencing research: A review of the literature, 1952-1994. *IEEE Transactions on Engineering Management*, 44, 316-329.

Reza Hejazi, S. and S. Saghafian (2005) Flowshop-scheduling problems with makespan criterion: a review. *International Journal of Production Research* 43, 2895-2929.

Ruiz, R. and Maroto, C. (2005). A comprehensive review and evaluation of permutation flow shop heuristics. *European Journal of Operational Research* 165, 479-494.

Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research* 47, 65-74.

Talwar, P.P. (1967). A Note on Sequencing Problems with Uncertain Job Times. *Journal of the Operations Research Society of Japan*, 9, 93-97.

Widmer, M. and A. Hertz (1989). A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research* 41, 186-193.

TABLE 1. Preliminary experiments with uniform distributions

machines	**2**		**jobs**		**10**			
Means	40-60		40-60		40-60		40-60	
Ranges	1		10		20		30	
Random	2.036%	0	2.751%	0	3.294%	0	2.965%	0
CDS/Johnson	0.000%	10	0.000%	8	0.017%	5	0.050%	6
CDS/Talwar	0.185%	4	0.257%	2	0.035%	4	0.086%	3
NEH	0.000%	9	0.149%	0	0.537%	0	0.557%	0
machines	**3**		**jobs**		**10**			
Random	3.805%	0	4.055%	0	3.681%	0	3.346%	0
CDS/Johnson	0.615%	3	0.661%	2	0.398%	1	0.280%	2
CDS/Talwar	0.531%	3	0.420%	0	0.245%	3	0.154%	4
NEH	0.008%	7	0.204%	4	0.327%	2	0.413%	1
machines	**6**		**jobs**		**10**			
Random	3.812%	0	3.558%	0	3.500%	0	3.312%	0
CDS/Johnson	0.316%	5	0.585%	2	0.284%	3	0.267%	5
CDS/Talwar	0.625%	0	0.823%	2	0.906%	0	0.458%	1
NEH	0.213%	4	0.267%	5	0.502%	3	0.362%	2

TABLE 2. Preliminary experiments with exponential distributions

machines	**2**		**jobs**		**10**			
Means	10-20		10-30		10-40		10-50	
Random	3.416%	0	5.587%	0	6.404%	0	6.957%	0
Johnson	0.269%	0	0.277%	0	0.164%	1	0.226%	1
Talwar	0.000%	10	0.000%	10	0.000%	10	0.000%	10
NEH	1.377%	0	1.886%	0	1.522%	0	0.240%	0
machines	**3**		**jobs**		**10**			
Random	3.783%	0	5.525%	0	7.285%	0	5.938%	0
Johnson	0.487%	0	0.453%	0	0.746%	1	0.583%	0
Talwar	0.000%	10	0.020%	9	0.023%	8	0.042%	9
NEH	0.942%	0	1.300%	1	1.194%	0	1.484%	1
machines	**6**		**jobs**		**10**			
Random	3.819%	0	5.657%	0	6.168%	0	11.020%	0
Johnson	0.233%	5	0.658%	5	0.929%	3	0.874%	2
Talwar	0.595%	2	0.869%	2	0.851%	3	0.566%	3
NEH	1.047%	0	1.239%	0	0.928%	4	0.911%	3

TABLE 3. Experiments with lognormal distributions and different standard deviations

machines	**2**		**jobs**		**10**					
Means	40-60		40-60		40-60		40-60		40-60	
σ	1		5		10		20		40	
Random	2.041%	0	2.268%	0	2.676%	0	2.574%	0	2.735%	0
Johnson	0.000%	10	0.002%	7	0.066%	5	0.111%	2	0.217%	1
Talwar	0.195%	4	0.087%	3	0.028%	5	0.013%	8	0.007%	9
NEH	0.017%	7	0.367%	2	0.742%	0	0.929%	0	0.919%	0
machines	**3**		**jobs**		**10**					
Random	3.250%	0	3.833%	0	3.569%	0	2.214%	0	2.737%	0
Johnson	0.645%	3	0.192%	4	0.199%	4	0.352%	1	0.283%	3
Talwar	0.699%	1	0.229%	2	0.178%	5	0.006%	9	0.027%	7
NEH	0.043%	6	0.352%	4	0.439%	1	0.623%	0	0.574%	0
machines	**6**		**jobs**		**10**					
Random	5.254%	0	3.086%	0	2.929%	0	3.152%	0	2.227%	0
Johnson	0.481%	1	0.298%	5	0.327%	2	0.318%	3	0.253%	5
Talwar	0.880%	1	0.449%	1	0.602%	2	0.205%	5	0.437%	3
NEH	0.043%	8	0.361%	4	0.383%	6	0.671%	1	0.410%	0

TABLE 4. Experiments with lognormal distributions and different means

machines	**2**		**jobs**		**10**					
Means	45-55		40-60		30-70		20-80		10-90	
σ	10		10		10		10		10	
Random	1.317%	0	2.315%	0	5.347%	0	6.921%	0	13.415%	0
Johnson	0.031%	2	0.104%	4	0.077%	7	0.025%	6	0.002%	5
Talwar	0.009%	8	0.031%	6	0.101%	2	0.163%	4	0.021%	5
NEH	0.414%	0	0.606%	0	0.651%	2	0.787%	1	0.425%	1
machines	**3**		**jobs**		**10**					
Random	2.039%	0	3.567%	0	6.640%	0	8.131%	0	15.792%	0
Johnson	0.136%	2	0.292%	2	0.684%	2	0.392%	3	1.127%	1
Talwar	0.011%	8	0.042%	6	0.540%	3	0.331%	5	0.195%	5
NEH	0.367%	0	0.565%	1	0.472%	5	0.469%	2	0.388%	4
machines	**6**		**jobs**		**10**					
Random	1.790%	0	2.760%	0	8.549%	0	12.872%	0	15.156%	0
Johnson	0.042%	8	0.263%	5	0.475%	2	1.289%	2	1.470%	1
Talwar	0.552%	0	0.405%	2	0.470%	2	2.202%	1	0.655%	2
NEH	0.406%	2	0.500%	2	0.200%	6	0.157%	7	0.353%	7

TABLE 5. Experiments with lognormal distributions and different means and standard deviations

machines	**2**		**jobs**		**10**					
Means	45-55		40-60		30-70		20-80		10-90	
σ	10-20		10-30		10-40		10-50		10-60	
Random	1.628%	0	3.868%	0	4.688%	0	5.755%	0	9.980%	0
Johnson	0.171%	1	0.106%	2	0.085%	4	0.181%	1	0.055%	2
Talwar	0.012%	9	0.047%	8	0.062%	5	0.000%	10	0.016%	8
NEH	0.742%	0	1.026%	1	1.303%	1	1.494%	0	1.019%	1
machines	**3**		**jobs**		**10**					
Random	1.382%	0	3.192%	0	5.702%	0	9.026%	0	10.984%	0
Johnson	0.161%	2	0.416%	0	0.684%	1	0.479%	2	1.012%	2
Talwar	0.014%	8	0.035%	9	0.025%	8	0.101%	5	0.100%	7
NEH	0.426%	0	0.652%	1	1.264%	0	1.187%	2	2.230%	1
machines	**6**		**jobs**		**10**					
Random	1.406%	0	3.263%	0	5.268%	0	7.670%	0	12.734%	0
Johnson	0.093%	5	0.568%	2	0.571%	3	0.512%	3	1.043%	3
Talwar	0.290%	2	0.115%	6	0.709%	0	0.356%	4	1.785%	0
NEH	0.245%	1	0.560%	2	0.192%	7	0.402%	3	0.184%	7