Research Notes for Chapter 10*

There is no question that Johnson's paper was pivotal in the development of scheduling theory. As a consequence the number of papers on flow shops is vast, and we can't begin to do justice to them all. For example, quite a few papers presented new proofs for Johnson's result. Nonetheless, we mention some historical results, and specifically we provide the sources we used in the chapter.

We observed a connection between the model with start- and stop lags and the model with separable setup. The time-lag model is due to Mitten (1959). Johnson (1959) commented and elaborated on the results. Among other things, Johnson provided a new proof based on his original treatment of three machines. Mitten considered only a start-lag but also required it to apply from the completion time on the first machine to the completion time on the second machine. That requirement follows directly from a standard assumption in the model, that once an operation starts on a machine it must be completed without interruption. Thus, an implicit stop lag of $v_j = u_j - A_j + B_j$ is also imposed. For a single job case, the makespan with a time lag is thus given by $\max\{u_j + A_j, u_j + B_j\}$. Define the *net lag* for the single-job case as $\max\{u_j + A_j, u_j + B_j\} - (A_j + B_j) = u_j - \min\{A_j + B_j\}$. When the net lag is positive, it increases the makespan; when it is negative, we obtain an overlap (in which case the makespan is decreased). Mitten's solution was by a generalization of Algorithm 10.2 that sorts the jobs in U by increasing u_j and the jobs in V by decreasing u_j. Therefore, if we were to set $u_j = \min\{A_j, B_j\}$—the zero net gap case—we would obtain Algorithm 10.2. Apparently, Mitten's algorithm is the origin of Algorithm 10.2 (whereas Algorithm 10.1 is due to Johnson himself). To wit, the following quote from Johnson (1959): "Note that Mitten's rule also applies to the standard two-stage problem in [Johnson 1954] and is perhaps easier to remember than the rule given in [Johnson 1954]." However, in (10.3) and (10.4) we did not follow the structure that we just described. Instead, we used a presentation that distinguishes explicitly between u_j and v_j, thus allowing them to be defined independently. For additional early references relating to this model, see Graham et al. (1979). Furthermore, recall that we described (10.3) and (10.4) as relevant to the best *permutation* sequence. The reason for that is simple: there are cases where the optimal solution is not a permutation. According to Graham, Lawler, Lenstra and Rinnooy Kan (1979), the general problem had been shown by Lenstra (unpublished) to be NP-hard in the strong sense. However, the best permutation cannot be suboptimal unless at least one job has a relatively large time lag. Johnson (1959) provided a sufficient condition that prohibits such an exchange: suppose jobs i and j are adjacent in that order on machine A, then they should not be considered for an interchange unless the net gap of job i exceeds

---

* The Research Notes series () accompanies our textbook *Principles of Sequencing and Scheduling,* Wiley (2009, 2019). The main purposes of the Research Notes series are to provide historical details about the development of sequencing and scheduling theory, expand the book's coverage for advanced readers, provide links to other relevant research, and identify important challenges and emerging research areas. Our coverage may be updated on an ongoing basis. We invite comments and corrections.

Citation details: Baker, K.R. and D. Trietsch (2019) Research Notes for Chapter 10 in *Principles of Sequencing and Scheduling* (Wiley, 2019). URL: http://faculty.tuck.dartmouth.edu/principles-sequencing-scheduling/.

that of job j plus $\max\{A_j, B_j\}$, which implies that such an interchange should certainly not be considered if the net gap of job i is negative; i.e., when the start gap reflects the use of transfer batches (with positive overlaps), the result is optimal.

The flow shop problem is NP-hard, in the strong sense, for $m \geq 3$. That result is due to Garey, Johnson and Sethi (1976). They also showed that minimizing F is NP-hard for $m \geq 2$ and if we allow recirculation then the job shop makespan problem is NP-hard for $m \geq 2$.* Recirculation occurs when a job visits the same machine more than once. The job shop model we present in Chapter 14 does not include recirculation but it is still NP-hard for $m \geq 3$ because it generalizes the flow shop model. For $m = 2$ but without recirculation, Jackson (1956) showed that Johnson's algorithm can be extended very simply. There are four job types to consider: jobs that require just one machine are denoted a or b whereas jobs that require both machines are denoted ab or ba, as per their operations sequence. Jobs of type a (b) can be considered to have zero processing times on machine 2 (1), so we may limit our attention to ab and ba. We then schedule both types by Johnson's Rule but for ba we treat machine 2 as the first machine. The actual machine loading then loads type ab on machine 1 in parallel to type ba on machine 2 and the operations of each type on its second machine are scheduled after the other type is complete. In such a schedule jobs of type a and b are scheduled in between the other two types but their internal order is immaterial (by Johnson's algorithm they are last on their machine because they require zero time on the other machine). Thus machine 1 is loaded $\{ab\}$-$\{a\}$-$\{ba\}$ and machine 2 is loaded $\{ba\}$-$\{b\}$-$\{ab\}$; both machines start processing at time zero.

In general, complexity proofs of the type we discuss here assume n jobs, where the data describing those jobs is part of the input. Nonetheless, we can also study problems with few jobs and many machines. One might think, for example, that a two-job m-machine flow shop is symmetric to an m-job two-machine flow shop, but in fact the problem is different. Specifically, the optimal solution may involve passing whereas the two-machine problem is optimized by a permutation schedule. The problem is still polynomial, however: even the more general m-machine, two-job job shop can be solved in polynomial time. Akers (1956) presented a graphical approach to the job shop problem—as posed by Akers and Friedman (1955)—and showed that it amounts to the solution of a shortest route problem. Szwarc (1960) showed in detail that this shortest path problem is indeed subject to efficient solution by dynamic programming but he, too, noted that it is easy to solve by inspection.† To appreciate the value of such an approach recall that at the time, computers were still at a nascent stage, and it would be very uneconomical to use them for a model that could be solved with paper and pencil. For instance, to describe the practicality of his rule, Johnson (1954) said: "For example, the decision rule permits one optimally to arrange twenty production items in about five minutes by visual inspection." Finally, Conway, Maxwell and Miller (1967) point out that the Akers' graphical approach can also handle recirculation. By contrast, recall that the n-job two-machine job shop becomes strongly NP-hard once we allow recirculation.

---

* Graham, Lawler, Lenstra and Rinnooy Kan (1979)—generally an excellent source for numerous important early publications and especially ones related to complexity—cite other sources for these results, but with later publication dates.

† The efficient solution of the shortest route problem—which we already described in our Research Notes for Chapter 8—was not yet well known at the time these papers were written. In response, Akers simply assumed the shortest route can be identified visually. Szwarc solved that problem more formally by a dynamic programming model that is essentially identical to Dijkstra's algorithm.

The special 3-machine cases of Section 10.4, starting with Johnson's own special case solution (when machine 2 is dominated), are discussed by Burns and Rooker (1976, 1978) and by Szwarc (1977). The test results we reported in the chapter are due to Smits and Baker (1981).

As mentioned in the chapter, although permutation schedules are not a dominant set for makespan problems when $m \geq 4$, it seems plausible that the best permutation schedule should be close to the optimum. That the worst case behavior of permutation schedules is not even bounded by a constant but may be roughly as large as $0.5m^{0.5}$ is due to Potts, Shmoys and Williamson (1991). Their example is based on an instance with fewer jobs than machines. Thus it does not contradict the other result we reported, due to Portougal and Scott (2001), who show that permutation schedules are asymptotically optimal for large n/m, both for minimizing the makespan and maximal tardiness.

Much of the early history of exact solutions for the problem is summarized in Baker (1975). The state of the art at the time did not include much testing and virtually no comparative testing of various solution ideas that had been published by then. Baker performed the necessary tests and found that some bounds and elimination methods did not justify the effort required for their calculation. As a footnote, it is much tougher to publish untested results today, but there is a simple explanation: at the time computer time was limited. For the same reason, Baker did not attempt to solve problems with more than 7 jobs on 7 machines. Later, a more advanced use of branch and bound was described by Potts (1980), whose approach uses a version of LB_2 and exploits considerations of symmetry in the branching process. His results indicate that problems containing 20 jobs and 4 machines may be considered "large," although they could be solved in a few seconds of computer time. Problem sizes of up to 50 or 100 jobs and 5 machines could often be solved with modest computational times. However, some problems of this size required very extensive searching. In the Potts study, for example, the algorithm was terminated in as many as 25% of the test problems with 50 or more jobs after generating 100,000 nodes in the branching tree. Finally, in Chapter 14 we discuss the shifting bottleneck algorithm, which constituted a breakthrough in the optimization of job shops. Here we may note that there is a strong similarity between that algorithm and the focus on one or two bottleneck machines that we describe in Chapter 10. In the shifting bottleneck algorithm, however, we may redefine the bottleneck machine as we go along.

The slope index algorithm is due to Palmer. In addition to the slope index method and the CDS method, which are both specialized to the flow shop model, we can use general-purpose heuristic techniques. For instance, Nawaz, Enscore and Ham (1983) apply the insertion heuristic to the flow shop problem, and their computational experience suggests that although the insertion heuristic requires more effort than the CDS algorithm, the additional effort produces slightly better solutions. Similarly, neighborhood search techniques have been explored by Dannenbring (1977) and Ho and Chang (1991), simulated annealing by Osman and Potts (1989), and tabu search by Taillard (1990), Reeves (1993), and Moccellin (1995). Heuristic procedures appear to be able to generate solutions that are within about 1% of optimum, on average. Worst-case analyses have been summarized by Lai (1996).

The source of our application of TSP to the solution of the no-wait flow shop is due to Wismer (1972), although the title and some of the narrative may be misinterpreted to suggest that the paper addresses the blocking problem. The narrative also suggests that the same approach could transform the job shop version of the same problem to a TSP, but there is no example to substantiate that claim. We refer to Bagchi, Gupta and Sriskandarajah (2006) for a more general survey of flow shop problems that can be addressed by the TSP. They also summarize a more modern rendition of the Gilmore and Gomory algorithm that runs in $O(n \log n)$ instead of the original $O(n^2)$. Last but not least, they cite a 1960 source for the application of TSP to the no-wait flow shop: Piehler (1960). Perhaps due to the fact that it's in German, that contribution was not well known in the US.

Studies of criteria other than the makespan for flow shop problems include Ho and Chang (1991) and Rajendran and Chaudhuri (1991) on the F-problem, Miyazaki and Nishiyama (1980) on the F_w-problem, and Kim (1993) and Raman (1995) on the T-problem.

References

Akers, S.B. and J. Friedman (1955) "A Non-Numerical Approach to Production Scheduling Problems," *Operations Research* 3, 429-442.

Akers, S.B. (1956) "A Graphical Approach to Production Scheduling Problems," *Operations Research* 4, 244-245.

Bagchi, T.P., J.N.D. Gupta and C. Sriskandarajah (2006) "A review of TSP based Approaches for flowshop scheduling," *European Journal of Operational Research* 169, 816–854.

Baker, K.R. (1975) "A Comparative Study of Flow-Shop Algorithms," *Operations Research* 23, 62-73.

Burns, F. and Rooker, J. (1976) "Johnson's Three-Machine Flow Shop Conjecture," *Operations Research* 24, 578-580.

Burns, F. and Rooker, J. (1978) "Three Stage Flow Shops with Regressive Second Stage," *Operations Research* 26, 207-208.

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.

Conway, R.W., W.L. Maxwell and L.W. Miller (1967) *Theory of Scheduling*, Addison-Wesley, Reading, MA.

Dannenbring, D. (1977) "An Evaluation of Flow Shop Sequencing Heuristics," *Management Science* 23, 1174-1182.

Garey, M.R., D.S. Johnson and R. Sethi (1976) "The Complexity of Flowshop and Jobshop Scheduling," *Mathematics of Operations Research* 1, 117-129.

Gilmore, P.C. and R.E. Gomory (1964) Scheduling a One-State Variable Machine: A Solvable Case of the Traveling Salesman Problem, *Operations Research* 12, 655–679.

Graham, R.L., E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan (1979) "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey," *Annals of Discrete Mathematics* 5, 287-326.

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.

Ignall, E. and L.E. Schrage (1965) "Application of the Branch and Bound Technique to Some Flow Shop Scheduling Problems," *Operations Research* 13, 400-412.

Jackson, J.R. (1956) "An Extension of Johnson's Results on Job Lot Scheduling," *Naval Research Logistics Quarterly* 3, 201-203.

Johnson, S.M. (1954) "Optimal Two-and Three-Stage Production Schedules with Setup Times Included," *Naval Research Logistics Quarterly* 1, 61-68.

Johnson, S.M. (1959) "Discussion: Sequencing *n* Jobs on Two Machines with Arbitrary Time Lags," *Management Science* 5, 299-303.

Kim, Y.D. (1993) "Heuristics for Flowshop Scheduling Problems Minimizing Mean Tardiness," *Journal of the Operational Research Society* 44, 19-28.

Mitten, L.G. (1959) "Sequencing *n* Jobs on Two Machines with Arbitrary Time Lags," *Management Science* 5, 293-298.

Lageweg, B.J., J.K. Lenstra and A.H.G. Rinnooy Kan (1978) "A General Bounding Scheme for the Permutation Flow Shop," *Operations Research* 26, 53-67.

Lai, T.-C. (1996) "A Note on Heuristics of Flow Shop Scheduling," *Operations Research* 44, 648-652.

Miyazaki, S. and N. Nishiyama (1980) "Analysis for Minimizing Weighted Mean Flow-time in Flow-shop Scheduling," *Journal of the Operations Research Society of Japan* 23, 118-132.

Moccellin, J.V. (1995) "A New Heuristic for the Permutation Flow Shop Scheduling Problem," *Journal of the Operational Research Society* 46, 883-886.

Monma, C. and A.H.G. Rinnooy Kan (1983) "A Concise Survey of Efficiently Solvable Special Cases of the Permutation Flow-Shop Problem," *RAIRO Recherche Operationelle* 17, 105-119.

Nawaz, M., E. Enscore and I. Ham (1983) "A Heuristic Algorithm for the *m*-Machine *n*-Job Flow-shop Sequencing Problem," *Omega* 11, 91-95.

Nowicki, E. and C. Smutnicki (1989) "Worst-case Analysis of an Approximation Algorithm for Flow Shop Scheduling," *Operations Research Letters* 10, 473-480.

Osman, I.H. and C.N. Potts (1989) "Simulated Annealing for Permutation Flow-Shop Scheduling," *Omega* 17, 551-557.

Palmer, D.S. (1965) "Sequencing Jobs Through a Multi-Stage Process in the Minimum Total Time—A Quick Method of Obtaining a Near Optimum," *Operational Research Quarterly* 16, 101-106.

Panwalkar, S.S. and C.R. Woolam (1980) "Ordered Flow Shop Problems with No In-process Waiting: Further Results," *Journal of the Operational Research Society* 30, 1039-1043.

Piehler, J. (1960) "Ein beitrag zum Reinhenfolgeproblem," *Unternehmensforschung* 4, 138–142.

Potts, C.N. (1980) "An adaptive branching rule for the permutation flow-shop problem," *European Journal of Operational Research* 5, 19-25.

Potts, C.N., D.B. Shmoys and D.P. Williamson (1991) "Permutation vs. Non-permutation Flow Shop Schedules," *Operations Research Letters* 10, 281-284.

Rajendran, C. and D. Chaudhuri (1991) "An Efficient Heuristic Approach for the Scheduling of Jobs in a Flowshop," *European Journal of Operational Research* 61, 318-325.

Raman, N. (1995) "Minimum tardiness scheduling in flow shops: Construction and evaluation of alternative solution approaches," Journal of Operations Management 12, 131-151.

Rinnooy Kan A.H.G. (1976) *Machine Scheduling Problems: Classification, Complexity and Computations*, Nijhoff, The Hague.

Reeves, C.R. (1993) "Improving the Efficiency of Tabu Search for Machine Sequencing Problems," *Journal of the Operational Research Society* 44, 375-382.

Smith, M.L., S.S. Panwalkar and R.A. Dudek (1976) "Flow Shop Sequencing Problems with Ordered Processing Time Matrices: A General Case," *Naval Research Logistics Quarterly* 23, 481-486.

Smits, A.J.M. and K.R. Baker (1981) "An Experimental Investigation of the Occurrence of Special Cases in the Three-Machine Flowshop Problem," *International Journal of Production Research* 19, 737-741.

Szwarc, W. (1977) "Optimal Two Machine Orderings in the 3x*n* Flow Shop Problem," *Operations Research* 25, 70-77.

Szwarc, W. (1983) "Flow Shop Problems with Time Lags," *Management Science* 29, 477-481.

Taillard, E. (1990) "Some Efficient Heuristic Methods for the Flow Shop Sequencing Problem," *European Journal of Operational Research* 47, 65-74.

Wismer, D.A. (1972) "Solution of the Flow Shop Scheduling Problem with No Intermediate Queues," *Operations Research* 20, 689-697.