Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Technical Note—A Scheduling Problem Involving Sequence Dependent Changeover Times

A. G. Lockett, A. P. Muhlemann,

Technical Notes

A Scheduling Problem Involving Sequence Dependent Changeover Times

A. G. Lockett and A. P. Muhlemann
University of Manchester, Manchester, United Kingdom

(Received April 19, 1971)

This note discusses a problem of scheduling jobs on a machine using various machine tools in which considerable changing of the tools is necessary, and the changeover time depends critically on the previous jobs. It derives a branch-and-bound algorithm, which has been shown to be computationally restrictive at the present time. Various heuristic methods have been tried and the computational results are very promising.

IN RECENT YEARS, with the increasing emphasis being placed on plant utilization and the return on capital investment, much work is being done in the area of production scheduling. However, perhaps one of the most common assumptions made by the authors of papers on this subject is that the changeover time between jobs is independent of the processing sequence. This is particularly true of solutions that profess to find the best sequence with respect to some criterion. The classic example is the model developed by JOHNSON[8] to minimize the process time for the two-machine (and, under certain conditions, three-machine), N-job problem. Most of the branch-and-bound solutions also make this assumption; typical is the paper by IGNALL AND SCHRAGE.[7] One exception is a formulation by LANZENAUER,[11] which allows the problem to be solved using binary integer programming. Computation experience in this case is rather disappointing, however, and the largest problem reported solved has five jobs and five machines.

Although the assumption of sequence independence of changeover times is frequently made, WHITE[12] reports that this assumption does not hold in a significant number of actual problem situations. Moreover, the range of changeover times indicates that the assumption is not approximately valid, and in many environments changeover times account for a large proportion of the time jobs occupy a machine.

Generally, the results of the 'optimizing' models tend to be disappointing when applied to problems of a realistic size. This is confirmed by CAMBELL[4] and his colleagues. In their paper they develop a heuristic based on Johnson's algorithm and the results seem very promising.

THE PROBLEM

A MACHINE CONSISTS of a turret holding various machine tools. The positions of the tools on the turret are referred to as 'stations.' Each tool may fit only in a specified station and there are many more tools than stations. A job consists of an engineering component requiring various operations, e.g., drilling, punching, etc., and may use one of the tools from each station. The jobs differ considerably in their tool requirements and may need tools that are not presently on the turret. If this occurs, the tools in question have to be changed at a considerable manpower cost and delay, and hence throughput is reduced. (Each change takes the same amount of time.) The machine in question is working at capacity and a backlog of work is building up, and, therefore, a method of scheduling is required that will increase throughput, i.e., minimize changeover times. Hence, the objective of the problem has been taken to be that of minimizing the total number of tool changes.

Although it is simple enough to state the objective, a mathematical statement of the calculation of the total number of changeovers is far from simple. It is in the nature of the changeovers that this problem differs from those few published papers in this area. Generally, it is assumed that the changeover time between two

TABLE I

ILLUSTRATION OF THE CHANGEOVER CALCULATION FOR SIX JOBS ON A MACHINE WITH FOUR STATIONS

(Zero indicates no tool is required in a station.)

Job	Tools required	Tools on machine
1	(1, 4, 7, 8)	(1, 4, 7, 8)
2	(2, 0, 6, 9)	(2, 4, 6, 9)
3	(2, 0, 0, 0)	(2, 4, 6, 9)
4	(0, 5, 7, 9)	(2, 5, 7, 9)
5	(3, 0, 7, 8)	(3, 5, 7, 8)
6	(2, 4, 6, 8)	(2, 4, 6, 8)
Changeovers:		
First-order serial	2 0 2 2	
Higher-order serial	1 2 1 0	

jobs depends only on the two jobs—for example, in the paper by BURSTALL.[3] In this case, the changeovers are given by a square matrix. However, for the problem in question, the changeover for the next job depends not only on the previous job, but on jobs before it also, if some of the stations are empty.

The number of tool changes can be thought of as made up of two components: First-order-serial changeovers, which are due to the tool requirements of the previous job; higher-order-serial changeovers, which are due to tool requirements of jobs before the previous job because the previous job did not require tools in certain stations.

Table I shows an example. If each job had required a tool in every station, then the problem would have been of the travelling-salesman type.

Various methods of scheduling have been tried and the results reported below. In each case the static problem has been considered; i.e., given a series of jobs, what is the optimum schedule? In reality, the problem is dynamic and should include other factors, such as the cost of deriving a schedule, the cost of delaying a job etc..

and the solution should also incorporate some updating procedure indicating when a new schedule has to be derived. These dynamic aspects are under study and will be reported later.

MATHEMATICAL FORMULATION

Suppose in general there are m stations on the turret, and each may hold at most one of r different tools. Each job may now be thought of as an ordered m-tuple of the integers zero through r. This indicates the tool required in each station, zero indicating that a particular station is not required.

Hence a complete file of n jobs can be represented as the set:

$$V=\{\mathbf{v}_1, \cdots, \mathbf{v}_n\}, \qquad \mathbf{v}_i=(v_{i1}, v_{i2}, \cdots, v_{im}), \qquad v_{ij}\epsilon\{0, 1, 2, \cdots, r\}=N.$$

So v_{ij} is the tool required in the jth station for the ith job, and $\mathbf{v}_i \epsilon N^m$.

Definition. Given two jobs $\mathbf{v}_1$ and $\mathbf{v}_2$, then $\mathbf{v}_1$ is said to *dominate* $\mathbf{v}_2$ if and only if $v_{2j}=v_{1j}$ for all $v_{2j}\neq 0$.

Clearly, all dominated jobs may be ignored in the analysis. Another concept that enables the number of jobs to be reduced is amalgamation.

Definition. Given a pair of jobs $\mathbf{v}_1$ and $\mathbf{v}_2$ such that, if $v_{1j}=v_{2j}$ when $v_{1j}\neq 0$ and $v_{2j}\neq 0$, then $\mathbf{v}_3$ defined by

$$v_{3j}=\begin{cases} v_{2j}, & \text{if} \quad v_{2j}\neq 0,\\ v_{1j}, & \text{if} \quad v_{1j}\neq 0,\\ 0, & \text{otherwise,}\end{cases}$$

is called the *amalgamation* of $\mathbf{v}_1$ and $\mathbf{v}_2$.

Although the set of jobs obtained after removal of those dominated is independent of the order of removal, this is not true of amalgamation. Clearly this depends on the order in which jobs are considered for amalgamation. Various methods were tried, but tests indicated that none was significantly better than the others. As a result, 'first-come–first-served' amalgamation was adopted.

Before it is possible to define the objective mathematically, it is necessary to decide on the boundary conditions. It will be assumed that the first job to be processed is fixed and that, after processing the last job, the machine is set up as if it were to process the first job again.

Clearly, given a set of amalgamated jobs $T=\{\mathbf{t}_1, \cdots, \mathbf{t}_n\}$, and assuming an ordering such that $\mathbf{t}_1$ is the first job to be processed, the objective is to find a permutation $(i_1, i_2, \cdots, i_n)$ of the integers $\{1, \cdots, n\}$ such that the changeover for the sequence of jobs $\mathbf{t}_1, \mathbf{t}_{i_2}, \mathbf{t}_{i_3}, \cdots, \mathbf{t}_{i_n}, \mathbf{t}_1$ is minimal.

AN ALGORITHM

Given a set of n jobs $\{\mathbf{t}_1, \cdots, \mathbf{t}_n\}$, a partial changeover time matrix P may be defined by

$$[P]_{ij}=\sum_{k=1}^{k=m} f(t_{ik}, t_{jk}),$$

where

$$f(a,b)=\begin{cases}0, & \text{if} \quad a=0, \quad b=0, \quad \text{or} \quad a=b,\\ 1, & \text{otherwise.}\end{cases}$$

This matrix is simply the first-order changeovers and gives a lower bound on the changeovers caused by doing job i before job j. The matrix P is used as the basis for the branch-and-bound solution. The total changeovers for the sequence of jobs $(\boldsymbol{t}_1, \cdots, \boldsymbol{t}_n)$ can be thought of as made up of

$$\sum_{i=1}^{i=n} \sum_{k=1}^{k=m} f(t_{ik}, t_{i+1k}) + \Lambda,$$

if $\boldsymbol{t}_{n+1}=\boldsymbol{t}_1$, and Λ is the higher-order-serial changeovers, calculated by summing the occurrences of different tools separated by zeros, over all positions.

The method of solution is similar to the one used by LITTLE, MURTY, SWEENEY, AND KAREL[9] to solve the travelling-salesman problem. The bounds produced at each stage, however, are different because of the higher-order-serial changeovers.

At each stage, the first-order changeovers for the solution are obtained as the sum of the appropriate changeover-matrix elements, together with the higher-order-serial changeovers. There was no difficulty in calculating this, since it was a by-product of the routine that 'blocked any subtours.'

The algorithm was coded in Fortran. Testing showed that it was inefficient for all but the smallest-sized problems. The reason for this was twofold, the narrowness of the distribution of the changeovers and the ineffectiveness of the lower bounds. First-order-serial changeover accounted for approximately 55 per cent of the total changeover instances. Two courses of action seemed appropriate, to try to improve the bounds or to attempt to develop heuristics. Much work has been done on investigating bounding procedures for this type of problem (for example, *see* ASHOUR AND QURAISHI[1]), but not much success has been achieved. Recently HELD AND KARP[6] presented a method for solving the travelling-salesman problem. Although computation experience with this method seems promising, we think that little would be gained by applying it to this problem, since the modified bounds needed would still not be tight enough. Following the example of Cambell[4] and his colleagues, we decided to experiment with heuristics.

A HEURISTIC APPROACH

THE PROBLEM UNDER consideration is different from most published models in that the changeover times are sequence dependent. This means that heuristics comparable with the ones used to solve the travelling-salesman problem would seem more appropriate than those used for the flow-scheduling environment. A very good survey of the travelling-salesman problem is presented in a paper by BELLMORE AND NEMHAUSER.[2] Accordingly, various heuristics were tried, as described below, and applied to queues of jobs (known as job files) of differing sizes. A further difficulty is the criterion for evaluating the different rules.

Measurement of Performance

We decided that, if some idea of the nature of the distribution of the number of changeover instances was available for a fixed job file, this would be invaluable for evaluating the different heuristics. An attempt to generate this distribution analytically failed because of the nature of the problem. As a last resort, Monte Carlo simulation was employed. For a given job file, sequences were sampled at random and the associated number of changeovers calculated. A histogram was built up and, after the required number of samples had been taken, was tabulated, along with estimates of the mean and standard deviation. This was repeated for job files of real data of varying size.

For job files of size 40 and above (before amalgamation), it appears that the normal distribution is a good approximation to the number of changeovers. Accordingly, a chi-squared goodness-of-fit test was run on the data, and it indicated that there was no evidence to suggest the contrary. The distributions are presented in Fig. 1.

This discovery enabled a meaningful comparison to be made of the performance of the different heuristics, in terms of the probability that better sequences could be obtained randomly.

The Heuristics

Various heuristics seemed appropriate and five have been tried. They are:

1. *Random ordering.* This was included in an attempt to simulate the existing scheduling method, and for minor comparisons.

2. *Travelling salesman without backtracking.* This is the first of the heuristics derived from the branch-and-bound solution. It is based on the penalty matrix P whose elements were defined previously. Little's[9] algorithm is applied to this matrix and the first feasible solution obtained by branching to the right is taken as the ordering for the jobs.

3. *Optimal travelling-salesman solution for the changeover-time matrix.* This is the second heuristic based on the branch-and-bound solution. The sequence followed is the one that is the solution to the travelling-salesman problem with the penalty matrix.

4. *Minimize tool changes.* This takes the form of a loading rule, the next job to be processed is the one that requires the least number of tool changes.

5. *The closest unvisited city.* This is a somewhat simpler heuristic, again based on the penalty matrix P. Given a set of jobs $S_1, \cdots, S_n$ and a partial sequence $S' = (i_1, \cdots, i_q)$, $q<n$, the next job to be processed is job S_j, where

$$P_{i_q j} = \min\{P_{i_q r}: r \neq i_k,\ k=1, \cdots, q\}.$$

This heuristic is described in detail by CONWAY, MAXWELL AND MILLER.[5]

COMPUTATIONAL EXPERIENCE

The five scheduling heuristics were applied to problems having job files of various sizes, covering a representative sample of real-life situations. The results are presented in Table II. It can be seen that no solutions to the travelling-salesman heuristic were obtained because a limit of 30 minutes was applied to each computer run. (These were carried out on an ICL 1905F, approximately one sixth as power-

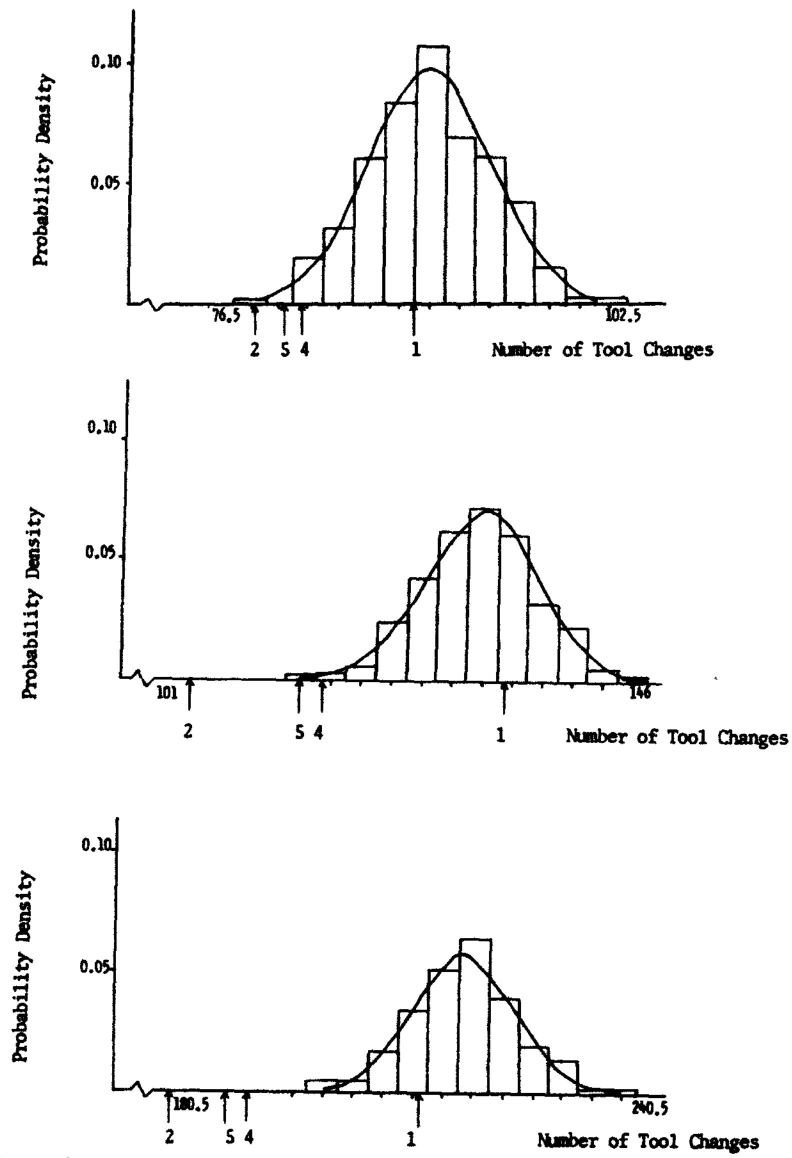

Fig. 1. Comparison of the heuristics and both sampled and hypothetical distributions. *Key:* 1. Random; 2. Travelling salesman without backtracking; 4. Minimize changeovers; 5. Closest unvisited city.

ful as an IBM 360/75.) All the other heuristics produced solutions in very short times and no difficulties were encountered. As a comparison, the results of the Monte Carlo simulation are presented. Examples of the distributions are given in Fig. 1.

TABLE II

RESULTS OF APPLYING THE HEURISTICS TO VARIOUS SETS OF JOBS

Number of jobs after amalgamation	Scheduling method—number of tool changes					Distribution	
	Random	Travelling salesman, no back-tracking	Travelling salesman, 'optimal'	Minimize change-overs	Closest unvisited city	Mean	Standard deviation
16	68	59	*	58	60	70.2	4.24
17	78	69	*	68	69	78.2	3.93
18	87	77	*	80	79	88.6	4.10
19	94	81	*	85	83	96.8	4.00
21	114	88	*	97	99	105.6	4.27
22	121	100	*	101	102	119.9	5.12
24	133	102	*	115	113	130.7	5.79
24	144	118	*	123	123	144.9	5.74
24	148	117	*	129	129	148.9	5.71
28	170	136	*	147	145	170.2	6.33
29	180	138	*	148	147	177.0	6.06
29	184	137	*	149	150	181.4	6.07
29	186	139	*	148	150	184.6	6.16
29	191	147	*	152	153	189.6	5.99
29	197	145	*	160	157	193.2	6.10
29	197	145	*	160	157	193.3	6.05
30	196	154	*	169	166	199.9	6.79
33	215	161	*	180	176	214.7	7.34
34	213	180	*	190	187	219.1	6.95
34	214	174	*	190	187	220.1	7.02
34	216	176	*	191	188	222.1	7.06
34	220	174	*	190	191	224.2	7.17
35	229	182	*	190	188	234.0	6.90
35	229	183	*	192	190	235.0	7.00

* No optimal solution found.

TABLE III

RESULTS OF THE SECOND HEURISTIC, EXPRESSED IN TERMS OF STANDARD DEVIATIONS FROM THE MEAN

Run no.	Heuristic 2 (no. of tool changes)	Standard deviations from mean	Run no.	Heuristic 2 (no. of tool changes)	Standard deviations from mean
1	59	2.64	13	139	7.40
2	69	2.34	14	147	7.11
3	77	2.83	15	145	7.90
4	81	3.95	16	145	7.98
5	88	4.12	17	154	6.96
6	100	3.87	18	161	7.32
7	102	4.69	19	180	5.63
8	118	4.69	20	174	6.57
9	117	5.59	21	176	6.53
10	136	5.40	22	174	7.00
11	138	6.44	23	182	7.54
12	137	7.32	24	183	7.43

An examination of Table II shows that, except for the first two smallest job files, the 'best' heuristic is number 2, i.e., travelling salesman, no backtracking. In general this heuristic produces solutions that are very far removed from the mean of the distribution. This is highlighted by reference to Fig. 1, and to Table III, which gives the solution measured in terms of the number of standard deviations from the mean.

ACKNOWLEDGMENT

THE AUTHORS are grateful to G. J. KYNCH for encouragement and advice during this research.

REFERENCES

1. S. ASHOUR AND M. N. QURAISHI, "Investigation of Various Bounding Procedures for Production-Scheduling Problems," *International J. of Production Res.* **7,** 249–252 (1969).
2. M. BELLMORE AND G. L. NEMHAUSER, "The Travelling Salesman Problem: A Survey," *Opns. Res.* **16,** 538–558 (1968).
3. R. M. BURSTALL, "A Heuristic Method for a Job Shop Scheduling Problem" *Opnal. Res. Quart.* **17,** 291–304 (1966).
4. H. G. CAMBELL, R. A. DUDEK, AND M. L. SMITH, "A Heuristic Algorithm for the *n* Job, *m* Machine Sequencing Problem", *Management Sci.* **16,** B630–637 (1970).
5. R. W. CONWAY, W. L. MAXWELL AND L. W. MILLER, *Theory of Scheduling,* Addison-Wesley, Reading, Mass., 1967.
6. M. HELD AND R. M. KARP, "A Combined Ascent and Branch and Bound Algorithm for the Travelling Salesman Problem," 7th Mathematical Programming Symposium (1970).
7. E. IGNALL AND L. SCHRAGE, "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," *Opns. Res.* **13,** 400–412 (1965).
8. S. M. JOHNSON, "Optimal Two and Three Stage Production Schedules with Set-Up Times Included," *Nav. Res. Log. Quart.* **1,** 61–68 (1954).
9. J. D. C. LITTLE, K. G. MURTY, D. W. SWEENEY, AND C. KAREL, "An Algorithm for the Travelling Salesman Problem," *Opns. Res.* **11,** 972–989 (1963).
10. E. USKUP, "Solution Methods for a Class of Two Stage Production Sequencing Problems," Doctoral dissertation, Illinois Institute of Technology, Chicago, Illinois (1970).
11. C. H. VON LANZENAUER, "A Production Scheduling Model by Bivalent Linear Programming," *Management Sci.* **17,** 105–111 (1970).
12. C. H. WHITE, "Sequence Dependent Set-up Times: A Prediction Method and an Associated Technique for Sequencing Production," Doctoral dissertation, University of Michigan (1966).