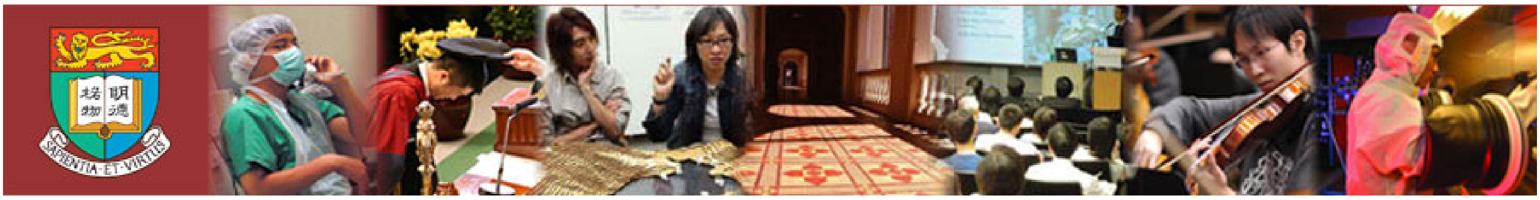


A sequencing algorithm for makespan minimization in FMS

Shiu-Hong Choi and James Siu-Lung Lee

The authors

Shiu-Hong Choi is Associate Professor and **James Siu-Lung Lee** is a research student, both in the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong.

Keywords

Sequential production, Tracking, Identification, Flexible manufacturing systems

Abstract

Job sequencing is an important stage in any hierarchical production control model, especially when a real-time dispatching rule is not employed. The problems become complicated when constraints, such as different parts requiring different operation processes at different machines and with different production priorities, are taken into consideration in the development of sequencing models. This paper first describes a mathematical programming model developed for small flexible manufacturing systems (FMS) makespan minimization sequencing problems. For large problems, a heuristic decomposition-based algorithm is proposed. The algorithm is based on the proposed concept of track generation and track identification. Each part type may require a different number of operation processes at different machines. A numerical example is used to illustrate the use of the algorithm.

Electronic access

The Emerald Research Register for this journal is available at **www.emeraldinsight.com/researchregister**

The current issue and full text archive of this journal is available at **www.emeraldinsight.com/1741-038X.htm**

Journal of Manufacturing Technology Management
Volume 15 · Number 3 · 2004 · pp. 291-297
© Emerald Group Publishing Limited · ISSN 1741-038X
DOI 10.1108/17410380410523515

Introduction

While research has been continually devoted to developing more efficient optimizing algorithms and faster and more economical computers, the present industrial scheduling problems can only be solved using heuristic techniques, which allow "good" solutions to be obtained. For the flowshop makespan scheduling problem, a number of heuristic algorithms are available, which provide quick solutions for different system configurations (Campbell *et al.*, 1970; Gupta, 1972; Page, 1961; Palmer, 1965).

An FMS can be considered as an automated job shop. The minimum makespan problem of job shop scheduling is a classical combinatorial optimization problem and only a few particular cases can be efficiently solved (Blazewicz *et al.*, 1996). Slight modifications of the problems turn out to be difficult. The job shop problems with two and three machines and operation-processing times equal to one or two, and equal to one, respectively, are NP-hard even in the case of pre-emption (Lenstra and Rinnooy Kan, 1979). Hefetz and Adiri (1982) and Kubiak and Timkovsky (1996) proposed algorithms for a two-machine case with unit-processing times. Hierarchical approaches for scheduling and control of flexible manufacturing systems (FMS) have been suggested and algorithms developed by Hildebrant (1980) and Buzacott and Yao (1986). The simple rule (shortest remaining first SRF schedule) proposed by Kubiak and Timkovsky (1996) states that the job with the smallest number of operations left should be scheduled next. Brucker (1988) used the network construction approach to solve the two-job-and-multi-machine case. The whole idea is to develop rules for constructing the shortest path in a two-job-and-multi-machine diagram first proposed by Akers (1956). Andreatta *et al.* (1995) proposed comparative equations to determine the maximum number of parts of a part type that may be processed consecutively for the minimum makespan in the two-machine case. A buffer system consisting of

Received: January 2001
Revised: June 2002
Accepted: July 2002
The authors would like to thank the Research Grant Council of the Hong Kong SAR Government for its financial support for this project.

two separate storage areas is situated between the two machines. Each storage area can only hold a stack of parts of the same part type. Unidirectional workflow is assumed in the study. There are some important points in the investigation of FMS scheduling problems. These include:

- makespan can be decreased by increasing the routeing flexibility; and
- the gain from increasing the routeing flexibility is highest when the number of different part types is limited.

A two-phase optimization method for multi-job-and-multi-machine problems was proposed by Lourenco (1995). The first phase consists of a large optimized transition in the current solution, while the second phase is basically a local search method. Although the results showed that such a large-step optimization method outperforms the simulated annealing method and finds the optimal schedules more frequently, the computational power required is extremely large and hence it is not practicable for solving problems in the industrial environment.

In this paper, a mathematical model is first formulated for small J-part-and-M-machine problems. For large FMS sequencing problems, a heuristic algorithm based on decomposition using track generation and identification is proposed. Each part type may require different operations, which are processed on different machines in random order. The proposed sequencing algorithms significantly reduce the number of schedules for consideration, and hence the computational power required. A numerical example is used to illustrate the use of the algorithm. The effectiveness of the proposed algorithm is demonstrated by comparing it with the results obtained from the mathematical model.

Formulation of the mathematical model

A mixed linear programming model is formulated to determine the sequence of jobs, so that the makespan is the minimum possible. It has the following features:

Parameters

j: index of part, where $j = 1, \ldots, J$;
p: index of operation, where $p = 1, \ldots, P$;
m: index of machine, where $m = 1, \ldots, M$;
L: any arbitrary large number;
μ_{jpm}: unit processing time of operation p of part j at machine m;

Control variables

t_{jpm}: completion time of operation p of part j at machine m;
Y_{ij}: equals 1 if the last operation of part i is finished before the last operation of part j, and 0 otherwise;
X^{m}_{jpik}: equals 1 if, at machine m, operation p of part j is finished before operation k of part i, and 0 otherwise;
C_j: equals t_{jpm} if part j is the last job to be finished in the schedule, and 0 otherwise.

Objective function

$$\min\left\{\sum_{j=1}^{J} C_j\right\}, \qquad (1)$$

subject to:

$$L \times (1 - Y_{ij}) + t_{jPm} - t_{iPm} \geq 0, \qquad (2)$$

$$L \times Y_{ij} + t_{iPm} - t_{jPm} > 0. \qquad (3)$$

Constraints (2) and (3) are used to determine which parts are completed first.

$$\sum_{i=j+1}^{J} (L \times Y_{ji}) + \sum_{i=1}^{j-1} [L \times (1 - Y_{ij})] + C_j \geq t_{jPm}. \qquad (4)$$

Equation (4) ensures that C_j equals t_{jPm}, if part j is the last job to be finished in the schedule:

$$t_{jPm} - t_{j(p-1)n} \geq \mu_{jPm}. \qquad (5)$$

The difference between the completion time of operations p and $p - 1$ of part j must be greater than or equal to the required processing time of operation p of that part.

Equation (6) ensures that the completion time of the first operation of part j is greater than or equal to the required processing time for that operation and part at machine m:

$$t_{j1m} \geq \mu_{j1m}. \qquad (6)$$

Constraints (7) and (8) ensure that no two different parts can be processed at the same machine simultaneously:

$$t_{jpm} - t_{ikm} + L \times X^{m}_{jpik} \geq t_{jpm}, \tag{7}$$

$$t_{ikm} - t_{jpm} + L \times (1 - X^{m}_{jpik}) \geq t_{ikm}. \tag{8}$$

Equations (9) and (10) restrict Y_{ij} and X^{m}_{jpik} to be zero-one decision variables:

$$Y_{ij} = 0, 1, \tag{9}$$

$$X^{m}_{jpik} = 0, 1. \tag{10}$$

There are two major assumptions made in the development of this model. First, the model assumes that there are sufficient material-handling devices to do all the handling jobs in time, so that no delay in loading and unloading of parts results. Second, the model assumes that the set-up time when switching from processing one part to another is zero. Since the set-up time of parts in a flexible manufacturing system is consistent, the second assumption can be released by adding the part set-up time to the processing time. The first assumption would be more difficult to release because it involves scheduling material-handling devices such as automated guided vehicles. The model has been implemented in a *C* program. It allows formulation of equations and converts them into the standard data codes of "quant system" for calculation.

This model is very flexible and it gives optimal solutions for small problems. For large problems, the long computational time required may not be practicable in a real manufacturing environment. An alternative approach is used to seek a quick and "good" solution. In the following section, a heuristic decomposition-based algorithm for a large FMS is described.

Sequencing algorithm for large FMS

Except by developing a closed form solution or some generic- or probabilistic-based method, the only possible way to simplify a complicated sequencing problem is to break the case down into several stages. The proposed algorithm consists of three stages, namely:

(1) track generation and identification stage;
(2) initial positioning of operations stage; and
(3) re-positioning of operations stage.

The idea of track generation and identification is to divide a complete sequencing problem, based on the processing pattern of different part types, into a number of sub-sequencing problems, so as to keep the number of iterations to the minimum possible. The final schedule is formulated by the gradual expansion of a sub-schedule. The procedure of the sequencing algorithm is described below.

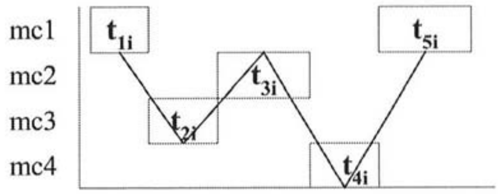
Step 1: For each part, construct a Gantt chart and connect the operations according to the production sequence by straight lines, as shown in Figure 1. t_{ji} represents the processing time of operation j of part i.

Step 2: For the part with the maximum number of straight lines in the Gantt chart, assign the first straight line as track 1, the second straight line as track 2, and so on. This part is termed the critical part, i_{crt}. In the case when there is more than one part having the maximum number of straight lines, choose any one of these parts as the critical part. Depending on the production process, the part with the maximum number of operations may not have the maximum number of straight lines.

Step 3: For part i (for $i \in I, i \neq i_{\text{crt}}$), if the gradient of its first straight line has the same sign as the first track of i_{crt}, i.e. having the same +ve gradient (sloping upwards) or −ve gradient (sloping downwards), assign its first straight line as track 1, the second straight line as track 2, and so on. Otherwise, assign the first straight line as track 2, the second straight line as track 3, and so on. Repeat Step 3 for all the parts except i_{crt}.

Step 4: Consider track k of all the parts. If it has a +ve gradient, the machine with the highest number in the track for all the parts is considered to be the first track machine of the track, and the second highest the second track machine, and so on. On the other hand, if track k has a −ve gradient, the machine with

Figure 1 Gantt chart of part type *i*



the smallest number in the track for all the parts is considered to be the first machine of the track, and the second lowest the second machine, and so on. If the last track machine in track k and the first track machine in track $k+1$ belong to the same machine, that machine is considered to belong to track k. The second track machine in track $k+1$ will automatically become the first track machine of track $k+1$, the third track machine in track $k+1$ becomes the second track machine of track $k+1$, and so on. Repeat Step 4 until all the tracks have been considered.

Step 5: Consider the operations that require the first track machine of track 1. If there is only one operation that requires the first track machine of track 1, assign that operation to the left of the Gantt chart. Otherwise, a higher priority is given to the one with a smaller processing time. The priority is arbitrary if the parts have the same processing time.

Step 6: Consider the operations that require the second track machine of track 1. If there is only one operation that requires that machine, position that operation to the left of the Gantt chart or behind the previously assigned operations, if any. Otherwise, operation j of part i has a lower production priority if the last assigned operation of part i has a later completion time compared with the last assigned operation of the other parts. Part i has the lowest priority if it has no operation assigned to the Gantt chart previously. If there is more than one part that does not have any operation assigned before, a higher priority is given to the one with a smaller processing time. Repeat Step 6 until all operations in the track have been assigned.

Step 7: Consider the operation with the latest completion time in the Gantt chart. Shift it to the left as far as possible, provided that the makespan is not increased. The operation is also shifted even if the makespan cannot be improved but remains the same. Shifting the operations is not restricted by the operations of other parts but by the precedence operation of the same part. Do not shift the operation if the resulting Gantt chart format is the same as the previous one. Repeat Step 8 until no further shifting of the last operation can result in a shorter or the same makespan.

Step 8: Repeat Steps 5 to 7 for the second track, then the third track until all the tracks have been considered.

Steps 1 to 4 are the track generation and identification stage. Steps 5 and 6 are the stage of part positioning, and Step 7 is the stage of re-positioning of parts. To illustrate the effectiveness of the proposed algorithm, a three-part-and-four-machine case is used as an example. The production requirement of each part is shown in Table I.

Solution

Steps 1-3: Construct a Gantt chart for each part and connect the operations according to the production sequence. Part A is the critical part, since it has the maximum number of straight lines. The track numbers for each part are assigned, as shown in Figure 2.

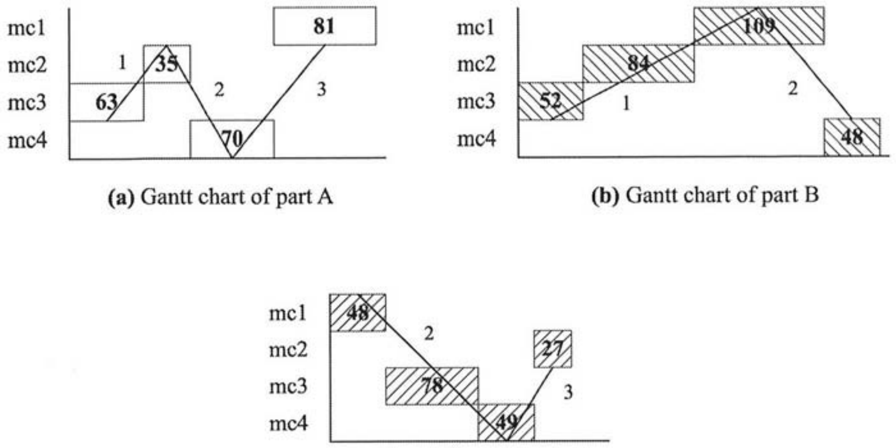
Step 4: Since track 1 has a +ve gradient, the first track machine is the one with the highest machine number, i.e. mc3. Therefore, the first track machine of track 1 is mc3, the second track machine is mc2, and the third track machine is mc1. For track 2, since it has a −ve gradient, the first track machine of track 2 is mc2, the second track machine is mc3, and the third track machine is mc4. The track machines in each track are listed in Table II.

Step 5: Consider the operations that require the first track machine (mc3) of track 1, i.e. operation 1 of parts A and B. This indicates that parts A and B compete for the same machine (mc3). A higher priority is therefore given to the

Table I Requirement of part types

Part	Operation number	Processing machine	Processing time in seconds
A	1	mc3	63
	2	mc2	35
	3	mc4	70
	4	mc1	81
B	1	mc3	52
	2	mc2	84
	3	mc1	109
	4	mc4	48
C	1	mc1	48
	2	mc3	78
	3	mc4	49
	4	mc2	27

Figure 2 Gantt charts of parts A, B and C



(a) Gantt chart of part A

(b) Gantt chart of part B

(c) Gantt chart of part C

Table II Machines in each track number

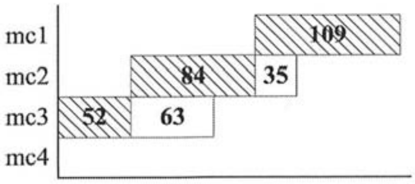
		Track machine	
Track number	**1st**	**2nd**	**3rd**
1	mc3	mc2	mc1
2	mc2	mc3	mc4
3	mc2	mc1	–

one with a smaller processing time, i.e. operation 1 of part B.

Step 6: Consider the operations that require the second track machine of track 1 (mc2), i.e. operation 2 of parts A and B. Since the last assigned operation of part A has a later completion time than that of the last assigned operation of part B, operation 2 of part B has a higher priority at the second machine of the track than that of operation 2 of part A.

Consider the operations that require the third track machine of track 1 (mc1), i.e. operation 3 of part B. Since there are no other operations from other parts that compete for the machine, assign that operation to machine 1. The complete assignment of the operations of the parts in track 1 is shown in Figure 3.

Figure 3 The Gantt chart for track 1



Step 7: Consider the operation with the latest completion time in the Gantt chart, i.e. operation 3 of part B. Shifting the operation to the left does not improve or maintain the performance. Then consider the operation with the latest completion time in the other machines, i.e. operation 2 of part A at machine 2. Again no better or the same performance results by shifting the operation to the left. Repeat the step until the operations with the latest completion time at all the machines have been tested.

Step 8: Go back to Step 5 for track 2.

Step 5: Consider the operations that require the first machine of track 2, i.e. operation 1 of part C. As there is only one operation in track 2 that requires mc1, assign that operation to mc1 posterior to all previously assigned operations at the same machine. Repeat Step 5 for the second machine of the track. For the third machine of the track, i.e. mc4, three operations, namely operations 3, 4 and three of part types A, B and C, respectively, compete for the same machine. Proceed to Step 6.

Step 6: The completion time of the last assigned operation of part A is less than that of the last assigned operation of part B, and the completion time of the last assigned operation of part B is less than that of the last assigned operation of part C. Therefore, operation 3 of part A has a higher production priority than that of operation 4 of part B at mc4, and operation 4 of part B has a higher production priority than that of operation 3 of part C at mc4. Figure 4 shows the Gantt chart for tracks 1 and 2.

Step 7: Consider the operation with the latest completion time, i.e. operation 3 of part C at mc4. Shifting the operation to the left does not improve or maintain the performance. Consider the operation with the latest completion time in the other machines, i.e. operation 2 of part C at mc3. Again no better or the same performance results by shifting the operation to the left. Consider the next operation with the latest completion time, i.e. operation 1 of part C at mc1. The performance is improved by shifting the operation to the left. The corresponding operations should also be moved to the left, subject to the operational precedence constraints, if the makespan can be improved or remain the same. Repeat Step 7 until the last operations at all machines have been tested. The sequence of the operations is arranged, as shown in Figure 5.

Step 8: Go back to Step 5 for track 3.

Step 5: Since there is only one operation that is required at each machine in track 3, operations are assigned to each of the corresponding machines without the need of determining the production priority.

Step 7: Re-positioning the operations with the latest completion time at each machine does not result in a better or the same performance. The problem is therefore finally scheduled, as shown in Figure 6.

The results of the decomposition-based algorithm and the mathematical model were compared with the optimal solutions obtained by the traditional full enumerative method. There was no obvious difference in these results, indicating that the performance of the proposed sequencing algorithms for minimization of makespan was satisfactory. However, there was a significant difference of the computational times of these approaches. For a 6-part-type-and-5-machine case on a PIII 650MHz PC, the sequence algorithm took less than ten seconds, and the programming model required about four minutes, while the traditional full enumerative method required a considerably longer time.

Figure 4 The Gantt chart for tracks 1 and 2

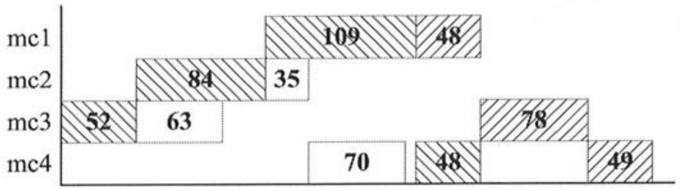
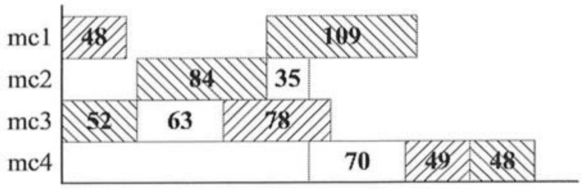


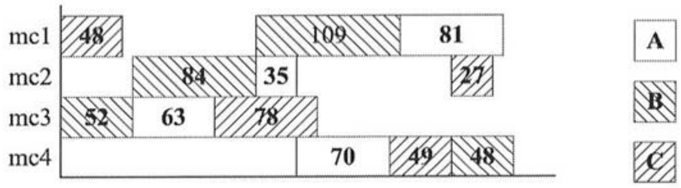
Figure 5 The Gantt chart for tracks 1 and 2 after repositioning



Conclusions

A mathematical programming model has been proposed for small J-part-type-and-M problems. For large FMS sequencing problems, a completely new concept of track generation and identification was proposed. The FMS sequencing algorithms requires formulation of sub-schedules and linkage of sub-schedules or operations in different tracks to form a complete schedule. The idea of the concept is to decompose a complex problem into a number of sub-sequencing problems based on the tracks identified for each part type. The algorithm also takes into account the effect of additional part types on sub-optimal schedules, hence ensuring that the final solution will not diverge much from the optimal solution. Re-positioning of operations must be done whenever there is a merge of new operations with sub-optimal schedules.

Figure 6 The final schedule for parts A, B and C



References

Akers, S.B. (1956), "A graphical approach to production-scheduling problems", *Operations Research*, Vol. 4, pp. 244-5.

Andreatta, G., Deserti, L. and Giraldo, L.N. (1995), "Scheduling algorithms for a two-machine flexible manufacturing system", *The International Journal of Flexible Manufacturing Systems*, Vol. 7, pp. 207-27.

Blazewicz, J., Domschke, W. and Pesch, E. (1996), "The job shop scheduling problem: conventional and new solution techniques", *European Journal of Operational Research*, Vol. 93, pp. 1-33.

Brucker, P. (1988), "An efficient algorithm for the job-shop problem with two jobs", *Computing*, Vol. 40, pp. 353-9.

Buzacott, J.A. and Yao, D.D. (1986), "Flexible manufacturing systems: a review of analytical models", *Journal of Management Science*, Vol. 32 No. 7, pp. 890-905.

Campbell, H.G., Dudek, R.A. and Smith, M.L. (1970), "A heuristic algorithm for the *n* job *m* machine sequencing problem", *Management Science*, Vol. 16, pp. B630-7.

Gupta, J.N.D. (1972), "Heuristic algorithms for multistage flowshop scheduling problem", *AIIE Transactions*, Vol. 4 No. 1, pp. 11-18.

Hefetz, N. and Adiri, I. (1982), "An efficient optimal algorithm for the two-machines unit-time job-shop schedule length problem", *Mathematics of Operations Research*, Vol. 7, pp. 354-60.

Hildebrant, R.R. (1980), "Scheduling flexible machining systems when machines are prone to failure", PhD thesis, MIT, Boston, MA.

Kubiak, W. and Timkovsky, V. (1996), "Total completion time minimization in two-machine job shops with unit-time operations", *European Journal of Operational Research*, Vol. 94, pp. 310-20.

Lenstra, J.K. and Rinnooy Kan, A.H.G. (1979), "Computational complexity of discrete optimisation problems", *Annals of Discrete Mathematics*, Vol. 4, pp. 121-40.

Lourenco, H.R. (1995), "Job-shop scheduling: a computational study of local search and large-step optimization methods", *European Journal of Operational Research*, Vol. 83, pp. 347-64.

Page, E.S. (1961), "An approach to the scheduling of jobs on machines", *Journal of Royal Statistical Society*, Vol. 23, Series B, pp. 484-92.

Palmer, D.S. (1965), "Sequencing jobs through a multistage process in the minimum total time – a quick method of obtaining a near optimum", *Operations Research Quarterly*, Vol. 16, pp. 101-7.

Further reading

Kimemia, J.G. (1982), "Hierarchical control of production in flexible manufacturing systems", PhD thesis, MIT, Boston, MA.