

ORDER RELEASE PLANNING IN A JOB SHOP USING A BI-DIRECTIONAL SIMULATION ALGORITHM

Chen-Tsau Chris Ying
Gordon M. Clark

Department of Industrial, Welding and Systems Engineering
The Ohio State University
Columbus, Ohio 43210, U.S.A.

ABSTRACT

Order release is an important job shop scheduling function that plans and controls the release of jobs to the shop floor. Deterministic simulation has been proposed to determine order release times with either a forward or a reversed approach. We develop a bi-directional algorithm that always starts and ends with a forward simulation run, and the algorithm includes a number of additional reversed and forward runs in between the first and last runs. A reversed simulation run determines potential job release times, and these potential times become actual release times when they are all non-negative. When one or more potential release times are negative, the algorithm modifies them to specify job release times for the succeeding forward simulation run. The last forward simulation run determines the job completion times.

Experimental results show that the bi-directional simulation algorithm produces a significantly improved mean flow time, and the algorithm can improve mean tardiness in some cases.

1 INTRODUCTION

Three essential elements of job shop scheduling are due-date assignment, order releasing, and job sequencing (Ahmed and Fisher 1992). Given a set of jobs with pre-determined job due dates, the order release plan has a significant effect on the performance of a job shop (Melnyk and Ragatz 1989, Ragatz and Mabert 1988). Ideally, the process producing the order release plan will consider the existing shop capacity and status, and the plan will avoid excessive queue times for jobs after they are released. Some authors (Melnyk and Ragatz 1989, Ragatz and Mabert 1988) believe that a good order release plan renders sequencing decisions less essential since the shop is less congested and the queue times of jobs are reduced. Obviously, no sequencing decisions are required in the absence of queues.

The traditional MRP approach assumes shop lead times are constant, and order releases are backward

scheduled ignoring the current shop status and the capacity constraint. The consequence could be late jobs, missing due dates, and/or creating an excessive amount of Work-in-Process (WIP).

To remedy this MRP weakness, authors have proposed a number of finite capacity approaches to more accurately describe the shop capacity and lead time. The Pritsker Corporation (1989), Wyman (1991), and Rosenwinkle and Rogers (1993) are applying discrete event simulation to schedule production. One basic advantage of simulation is that it can accurately represent the capacity and current status of a plant. Muller, Jackson and Fitzwater (1990) propose the use of simulation for order release. However, an attempt by simulation to identify order release times that minimize WIP and avoid late jobs may require many simulation trials representing different possible release times. Another approach is finite capacity backward loading (Ragatz and Mabert 1988). Although this approach is intuitively appealing, experimental results in the literature do not show significant improvement over the infinite-capacity approach.

Several authors (Watson, Medeiros, and Sadowski 1993; Mejsky 1985) introduce the "backward simulation" approach as an alternative to forward simulation. In fact, these "backward simulation" approaches are actually a regular forward simulation performed on a set of reversed job routes. The basic idea is to start jobs at their due dates, represent reversed job routes, and use the job completion times as the order release times. We call this approach the reversed problem formulation instead of backward simulation since the simulation does not represent a model that has a corresponding representation when simulating forward in time. That is, the state transition function does not, in general, have an inverse defined for a forward simulation (Ying 1994).

1.1 Overview of the Bi-Directional Algorithm

In this paper, we present a bi-directional simulation algorithm to determine job release times given a set of common job due dates. Assume that the planning system

specifies order release times periodically, i.e., once each planning period such as a week. For example, the master schedule may require the jobs at the end of the current planning period. In addition, the algorithm must consider the effect of a set of open orders (old jobs) released during the previous planning period and not yet completed.

The algorithm starts with a forward simulation run considering all open orders and determines the completion time of each job. These completion times become the starting point of a subsequent reversed simulation run which is the second run of the algorithm. Thus, the second run has a feasible starting condition which is not necessarily true if the job due dates were used for the first reversed run. The completion times of the reversed run, i.e. the second run, become a set of job ready times. If a job's ready time is negative, the algorithm sets it to zero. A job ready time is the time a job is ready to enter the system. The algorithm uses these ready times to calculate job release times and as a starting condition for the subsequent forward simulation, i.e., the third run for the algorithm. This third run may terminate the algorithm, and, if so, this forward simulation generates a set of completion times for use in calculating flow time and tardiness performance measures. The algorithm may continue for five runs in which case the completion times of the third run become the starting condition for the following reversed simulation. The algorithm always ends with a forward simulation.

In the following sections, we present a more detailed description of the reversed problem formulation and the bi-directional algorithm. Then, the paper describes the experimental setting and results.

2 THE REVERSED PROBLEM FORMULATION

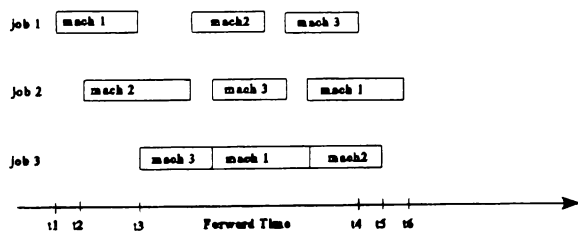


Figure 1: Forward Simulation

One can understand the role of the reversed problem formulation by considering the result of the preceding forward simulation. Figure 1 depicts a three-machine three-job forward simulation. Every job released in the job shop has an assigned due date and a route, which consists of a number of operations. Each operation specifies a machine and a processing time on that machine. This processing time is deterministic. The set of jobs waiting to be released to the shop floor is known. That is, we only

consider jobs available during the planning period. The times t_1 , t_2 and t_3 are release times, and the simulation computes the job completion times t_4 , t_5 and t_6 .

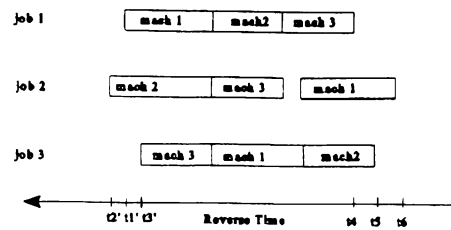


Figure 2: Reversed Simulation

The primary feature of the reversed problem formulation is to reverse the job route and the time axis. The reversed simulation starts with the job ready times equal to t_4 , t_5 , and t_6 as determined by the previous forward simulation. The job completion times for the reversed simulation give new values for t_1 , t_2 , and t_3 noted as t_1' , t_2' , and t_3' . Note the significant reduction in idle time for jobs 1 and 2 so $t_1' > t_1$ and $t_2' > t_2$, allowing for later release of jobs 1 and 2. When simulating backwards in time, idle time will disappear when both jobs and machines are simultaneously available allowing release times to be delayed.

A desirable characteristic of the reversed problem formulation is that it can be represented using software designed for forward simulation. If we truly attempt backwards simulation where the simulator traces events backwards in time for a model defined for a forward simulation, backwards simulation would require new software (Ying 1994). Consider, for example, the task of representing a backwards simulation of a job shop using a FIFO queue discipline when operating forwards in time. The following notation defines the initialization for a reversed simulation based on the result of a preceding forward simulation.

J is the number of jobs to be scheduled (includes both old and new jobs)

M is the set of all machines

N_j is the number of operations in job j 's route, $N_j > 0$; $j = 1, J$.

$P_{i,j}$ is the processing time of operation i on job j 's route in the forward simulation, $P_{i,j} > 0$; $i = 1, N_j$; $j = 1, J$.

$R_{i,j}$ is the machine for operation i on job j 's route for the forward simulation, $R_{i,j} \in M$; $i = 1, N_j$; $j = 1, J$.

$C(j)$ is the completion time of job j in the previous forward simulation; $j = 1, J$.

The following specifications define the reversed problem formulation for simulation using existing simulation software.

$R'_{i,j}$ is the machine for operation i on job j 's route in the reversed formulation, $R'_{i,j} \in M; i = 1, N_j; j = 1, J$.

$R'_{i,j} = R_{N_j-i+1,j}; i = 1, N_j; j = 1, J$.

$P'_{i,j}$ = processing time of operation i on job j 's route in the reversed formulation; $i = 1, N_j; j = 1, J$.

$P'_{i,j} = P_{N_j-i+1,j}; i = 1, N_j; j = 1, J$.

$r'(j)$ is the ready time of job j in the reversed setting.

$Cmax$ is the maximum job completion time, i.e. $Cmax = \max\{C(j), j = 1, J\}$.

$r'(j) = Cmax - C(j); j = 1, J$.

3 THE BI-DIRECTIONAL ALGORITHM

The bi-directional simulation algorithm consists of a sequence of forward and backward simulation runs. The algorithm starts with a forward simulation given a set of jobs to be released. We consider all open orders throughout the entire multi-pass simulation. The first forward simulation run generates a set of job completion times. A reversed simulation run follows starting with job completion times, which becomes job ready times in the reversed problem formulation. The reversed simulation then generates a new set of job ready times. If any job ready time is negative, the algorithm changes it to zero (the present time). The ready time for an old job is always zero for a forward simulation run. Another forward simulation run then follows starting with the revised set of job ready times. Note that job ready time is not necessarily the job's planned release time. If the job is able to access the first machine in its route, this job ready time becomes its release time. Otherwise, the release time becomes the time the job is able to access the first machine in its route. The forward-reversed-forward runs can be repeated as many times as desired.

The open orders, i.e., old jobs, merit special treatment in forward simulation runs. We make the assumption that the old jobs already have due dates assigned and/or they are already late. Thus, the algorithm gives the set of old jobs higher priority in any machine queue than any new jobs.

In calculating flow times based on the results of the last forward simulation, the algorithm implements the following rules.

- The flow time for a new job is the difference between its completion time and its release time.
- The flow time for an old job is its completion time.

4 EXPERIMENTAL SETTING

The experiments used a job-shop model described below to test the performance of the algorithm. The job shop consists of five machines and a set of jobs waiting to be released. The number of jobs in the system ranges from

25 to 250. Each new job has a route consisting of five operations. Any machine can be visited more than once by the same job as long as the machine is not required in two consecutive operations of the same job route.

The algorithm assumes all inputs, i.e., routes and processing times, are deterministic. To test the algorithm, the experiments used a problem generator that samples for processing times and routes to generate different problems. After one operation, a job has an equal probability of visiting any one of the other four machines. The problem generator determined processing times by sampling from a uniform distribution bounded by 1 and 10. The due date for new jobs assumes a 100-job shop load and 80% load/capacity ratio. Then we replicate scenarios with 25 jobs, 50 jobs, ..., and 250 jobs to represent 20%, 40%, ..., and 200% load/capacity ratios. If the sampled processing times did not give a load/capacity ratio within $\pm 5\%$ of the nominal 20%, 40%, ..., 200% values, the problem generator rejects that sample.

We use the Shortest-Processing-Time (SPT) rule as the sequencing rule in the algorithm for both forward and reversed runs. The SPT rule gives priority to the job with the shortest processing time of the competing jobs in any queue (with the exception that an old job has higher priority than any new job). The SPT rule has been tested by many researchers and shown to be effective in minimizing the mean job flow time (Baker 1984, Kiran and Smith 1984, and Dumond and Mabert 1988). We tested the FIFO rule in preliminary experiments not reported in this paper, and the results show the same pattern as produced by the SPT rule.

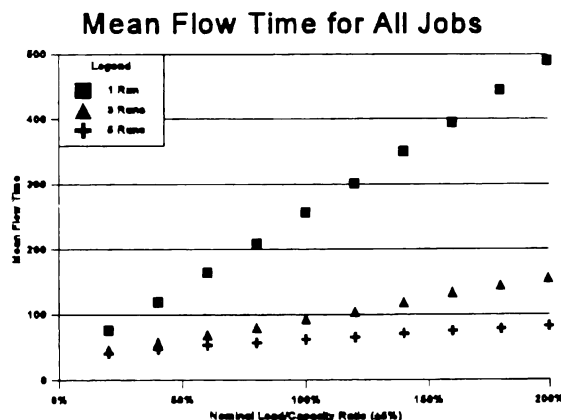


Figure 3: Mean Flow Time

The experiments use 10 % of all jobs to represent open orders and assume these jobs to have only 3 operations left in their routes at present time. The experimental procedure used a single forward simulation to determine their due dates. This simulation used the FIFO rule and did not include new jobs. The due dates for the

old jobs are the completion times generated by the simulation; thus, old jobs had tight due dates since they were based on no competition with new jobs.

5 EXPERIMENTAL RESULTS AND DISCUSSION

Figures 3 and 4 show the experimental results of this bi-directional simulation algorithm with both 3 and 5 runs. For comparison purposes, the figures include the results for a single forward simulation. Numbers in the figures are the average of 100 replications. The 90% confidence intervals half widths (CIHW) are all less than 1.6% of the mean values shown in Figure 3. In Figure 4, the CIHW values are less than 16% of the mean values for load/capacity ratios 20% and 40%, no more than 10% of the means for ratios 60%, 80%, and 100%, and less than 2.85% of the means for ratios greater than 100%. Figure 3 shows the mean flow time for all jobs aggregated together, i.e., old jobs and new jobs. Figure 4 shows the mean tardiness for all jobs.

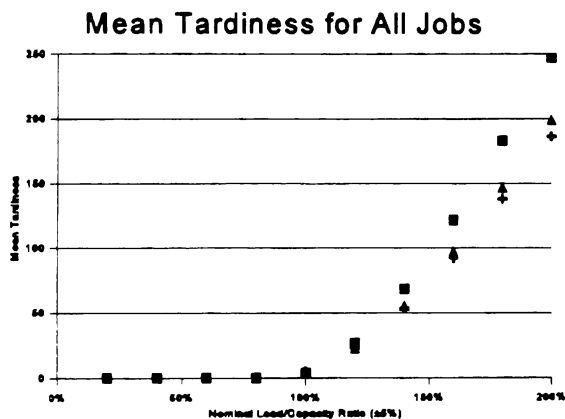


Figure 4: Mean Tardiness

The results show that the bi-directional simulation algorithm yields much better mean flow time over all jobs than a single forward simulation run. The improvement in the overall mean flow time ranges from about 40% in the 25-job case with 3 runs to more than 80% in the 200-job-or-more cases with 5 runs. While the overall mean flow time is improved significantly by the extra bi-directional simulation runs, the mean tardiness measure varies in different cases. The only case where the average tardiness increased is 100% load capacity ratio with 125 total jobs in the system. When the load-capacity ratio is less than 100%, there is no significant change in the overall mean tardiness caused by extra bi-directional simulation runs. When the load-capacity ratio is more than 100%, there is again a significant improvement by the extra runs. The improvement ranges from about 14% to 25%.

The above experiments tested the algorithm with

3 and 5 runs. The trend seems to indicate that each additional run will improve the mean flow time measure without significantly degrading mean tardiness. Finding the eventual lower bound for this approach and the number of runs to reach the lower bound is an interesting research question.

6 CONCLUSIONS

The experimental results show that bi-directional simulation is a promising approach for planning order release in a job shop. The results are dramatic since they show a significant reduction in mean flow time of jobs while still improving in most cases the mean tardiness. The improvement in overall mean flow time ranges from about 40% to more than 80% compared to a single forward run. When the shop load-capacity ratio is more than 100%, the algorithm improves the tardiness measure from 14% to 25%. When the shop load-capacity ratio is less than or equal to 100%, the results ranged from no significant effect of mean tardiness to an increase of 18%.

We believe that the ability of the approach to find late release times for jobs and avoid queuing accounts for the algorithm's superior performance.

REFERENCES

- Ahmed, Imtiaz and Warren W. Fisher. 1992. Due Date Assignment, Job Order Release, and Sequencing Interaction in Job Shop Scheduling. *Decision Sciences* 23:633-647.
- Baker, K. R. 1984. Sequencing Rules and Due-Date Assignments in a Job Shop. *Management Science*. 30: 1093-1104.
- Dumond, John and Vincent A. Mabert. 1988. Evaluating Project Scheduling and Due Date Assignment Procedures: An Experimental Analysis. *Management Science* 34:101-118.
- Kiran, A. S. and M. L. Smith. 1984. Simulation Studies in Job Shop Scheduling-I. *Comput. & Indus. Engng.* 8:87-93.
- Kiran, A. S. and M. L. Smith. 1984. Simulation Studies in Job Shop Scheduling-II. *Comput. & Indus. Engng.* 8: 95-105
- Melnik, Steven A. and Gary L. Ragatz. 1989. Order Review/ Release: Research and Perspectives. *Int. J. Production Research* 27:1081-1096.
- Mejtsky, G. J. 1985. Backward Simulation and Multiple-Objective Optimization of Job Shop Scheduling with Zero Tardiness and Minimum Makespan. In *Proceedings of the 1985 Summer Computer Simulation Conference*, 716-720.
- Muller, Daniel J., John K. Jackman, and John K. Fitzwater. 1990. A Simulation-Based Work Order Release Mechanism for a Flexible Manufacturing

- System. In *Proceedings of the 1990 Winter Simulation Conference*, ed. O. Balci, R.P. Sadowski, and R.E. Nance, 599 - 602. Institute of Electrical and Electronic Engineers, San Francisco, California.
- Pritsker Corporation. 1989. *FACTOR Implementation Guide*. West Lafayette, Indiana.
- Ragatz, Gary L. and Vincent A. Mabert. 1988. An Evaluation of Order Release Mechanisms in A Job-Shop Environment. *Decision Sciences* 19:167-189.
- Rosenwinkle, Maureen T. and Paul Rogers. 1993. Simulation-Based Finite Capacity Scheduling: A Case Study. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 939-946. Institute of Electrical and Electronics Engineers, Piscataway, NJ.
- Watson, Edward F., Deborah J. Medeiros, and Randall P. Sadowski. 1993. Generating Component Release Plans with Backward Simulation. In *Proceedings of 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 930-938. Institute of Electrical and Electronics Engineers, Piscataway, NJ.
- Wyman, F. Paul. 1991. Common Features of Simulation Based Scheduling. In *Proceedings of 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton and G.M. Clark, 341-347. Institute of Electrical and Electronics Engineers, Piscataway, NJ.
- Ying, Chen-Tsau. 1994. Backwards Simulation for Production Scheduling Purposes. Ph.D. Dissertation Draft. The Ohio State University, Columbus, OH.

AUTHOR BIOGRAPHIES

CHEN-TSAU CHRIS YING is a Ph.D. candidate in the Department of Industrial, Welding and Systems Engineering at The Ohio State University. He has a B.S.M.E. degree from Chung-Yuan Christian University and a M.S.I.E. degree from The Wichita State University. His current research interests include simulation-based production scheduling, statistical process control, and automation. He is a member of ORSA, TIMS, and IIE.

GORDON M. CLARK is a professor in the department of Industrial, Welding and Systems Engineering at The Ohio State University. He received the B.I.E. degree from the The Ohio State University in 1957, the M.Sc. in Industrial Engineering from The University of Southern California in 1965, and the Ph.D. degree from The Ohio State University in 1969. His current research interests include integrated manufacturing decision support systems in a CIM environment and the design and analysis of efficient simulation experiments. He is an associate editor for the *ORSA Journal on Computing*. In 1991, he served as program chair for the WSC.