International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: http://www.tandfonline.com/loi/tprs20

Satisfying due-dates in a job shop with sequence-dependent family set-ups

Mehmet R. Taner , Thom J. Hodgson , Russell E. King & Kristin A. Thoney

To cite this article: Mehmet R. Taner , Thom J. Hodgson , Russell E. King & Kristin A. Thoney (2003) Satisfying due-dates in a job shop with sequence-dependent family set-ups, International Journal of Production Research, 41:17, 4153-4169, DOI: 10.1080/0020754031000149167

To link to this article: http://dx.doi.org/10.1080/0020754031000149167

Published online: 14 Nov 2010.

INT. J. PROD. RES., 2003, VOL. 41, NO. 17, 4153–4169

Taylor & Francis
Taylor & Francis Group

Satisfying due-dates in a job shop with sequence-dependent family set-ups

MEHMET R. TANER†, THOM J. HODGSON‡,
RUSSELL E. KING‡* and KRISTIN A. THONEY§

This paper addresses job shop scheduling with sequence dependent family set-ups. Based on a simple, single-machine dynamic scheduling problem, state dependent scheduling rules for the single machine problem are developed and tested using Markov Decision Processes. Then, a generalized scheduling policy for the job shop problem is established based on a characterization of the optimal policy. The policy is combined with a 'forecasting' mechanism to utilize global shop floor information for local dispatching decisions. Computational results show that performance is significantly better than that of existing alternative policies.

1. Introduction

A job shop scheduling problem with sequence dependent family set-ups is considered. It is assumed that the jobs are released simultaneously. There are N families with n_i jobs in each family $i = 1,2,\ldots,N$. Each job j has a given due-date, d_j, and processing time, p_{jk}, on each machine, $k = 1,2,\ldots,M$. Set-up is required on some machines if a job from a smaller indexed family is an immediate successor of one from a larger indexed family. When required, the set-up time between two families is a given constant s. The initial set-up of each machine is assumed known. The objective is to determine a schedule that minimizes the maximum lateness ($L_{\max}$); that is, the maximum difference between the completion time and the due date, among all jobs.

The special set-up structure in this problem may be seen in process industries. In Conway *et al.* (1967) it is described as the *comedown problem*, which occurs in the rolling of steel strips. The rollers are slightly scored by the edges of the strip being rolled. Consequently, the next strip in the sequence must be narrower or the rollers must be reground. Another closely related problem is dyeing operations in the textile industry. Usually, a relatively minor set-up is involved when lighter to darker shades are dyed in progression. However, when a lighter shade is to be dyed following a darker shade, the dye vessel requires additional cleaning.

In the following, a brief overview of research on single-machine and job shop scheduling problems with sequence dependent set-up times and $L_{\max}$ performance criterion is presented. Then, a simple single machine problem is modelled and solved using Markov Decision Processes (MDP). A generalized scheduling policy

Revision received January 2003.

†Department of Industrial Engineering, Bilkent University, 06533 Ankara, Turkey.
‡Department of Industrial Engineering, North Carolina State University, Raleigh, NC 27695, USA.
§Department of Textile and Apparel, Technology and Management, North Carolina State University, Raleigh, NC 27695, USA.
*To whom correspondence should be addressed. e-mail: king@eos.ncsu.edu

International Journal of Production Research ISSN 0020–7543 print/ISSN 1366–588X online
http://www.tandf.co.uk/journals
DOI: 10.1080/0020754031000149167

is developed based on the optimal results from the MDP. This generalized policy is incorporated in the Virtual Factory (Hodgson *et al.* 1998, 2000) a job shop scheduling system. Experimentation is performed with different job shop scenarios, and results are compared with those obtained from alternative techniques.

2. Previous research

Many studies have appeared in the literature on general job shop scheduling problems. Pinedo (1995) gives a good summary of various approaches. A review of research on scheduling in the presence of set-up times can be found in Allahverdi *et al.* (1999).

Although there are a number of studies on single-machine problems with sequence dependent set-up times and the $L_{\max}$ performance criterion, we found no reported results in job shop scheduling with the same criterion except for those by Uzsoy and his colleagues. Ovacik and Uzsoy (1994a) use deterministic simulation to predict the arrival times of jobs at a machine in a job shop environment. They then utilize that information to make scheduling decisions whenever the machine becomes available. Their procedure is similar to dispatching but they also consider jobs that would arrive within a certain time window. Ovacik and Uzsoy (1994b) introduce rolling horizon algorithms to solve the single machine problems that occur as sub-problems in the decomposition of complex job shop scenarios. They develop a branch and bound procedure for small problems.

Uzsoy *et al.* (1991) develop a procedure that decomposes a complex job shop with sequence dependent set-up times into separate work centres. The job shop is represented as a disjunctive graph and the release times and due-dates of the jobs at each work centre are estimated. A separate scheduling problem is solved for each work centre. One of the arising problems is the single-machine scheduling problem under precedence constraints. They propose a branch and bound algorithm and a local improvement heuristic for the $L_{\max}$ objective, and develop an approximate procedure to minimize the number of tardy jobs. Uzsoy *et al.* (1992) propose heuristics for the dynamic problem and a dynamic programming procedure for the static problem to minimize $L_{\max}$ and the number of tardy jobs. They develop worst-case error bounds for the special case in which the set-up times are bounded from above by the processing times. Monma and Potts (1989) consider batch-scheduling problems in which the number of batches is fixed and the triangular inequality of set-ups is satisfied. They show that under those conditions the problems of maximum lateness, total weighted completion time and number of tardy jobs can be efficiently solved via dynamic programming.

To the best of our knowledge, this is the first paper in the literature on job shop scheduling problems where sequential processing of jobs belonging to different families involves sequence dependent set-up times.

3. A dynamic, single machine model

In this section, a single-machine, dynamic scheduling problem with due dates and sequence dependent set-up times is studied. A queuing model is developed. Optimal control rules to minimize a penalty cost are established using Markov Decision Processes. Since the objective value is additive in the MDP, using a minimax objective is difficult. However, the penalty cost structure used does approximate minimizing $L_{\max}$. The control rules obtained from the MDP are generalized and used to construct a heuristic scheduling policy that is applied

to single-machine sub-problems arising in the context of scheduling a complex job shop. This heuristic policy is tested against other rules that minimize $L_{\max}$ using a job shop simulation. This approach is similar to those of Matsui and Shinghu (1978) and Hodgson *et al.* (1987) although their approaches were applied to different problems.

3.1. *Problem definition*

A simple problem with three job families is considered. If the next job to be processed is from the same or higher indexed family as the previous job, then no set-up is required. However, if it has a lower family index, then a set-up in incurred. Thus, a job from family three never requires a set-up, while one from family two only requires a set-up if it follows a family three job and a family one job incurs a set-up unless it follows another family one job. The underlying probability distribution of the arrival process is Poisson, and the set-up and processing times are known and deterministic. An arriving job can be from family one, two, or three with given probabilities, r_1, r_2, and r_3, respectively. The time from the arrival of a job until its due date is a pre-specified constant, d.

After a job's lateness reaches a pre-specified point, denoted by l, a penalty cost is assessed for each additional period until the job is completed. The objective is to determine a schedule that minimizes the total penalty cost. This is equivalent to minimizing the number of periods that any job's lateness exceeds l time units. Hence, this objective also does an excellent job of minimizing $L_{\max}$ when the minimum $L_{\max}$ value exceeds l. The performance criterion implies the use of due date related and set-up time related scheduling rules, or a hybrid of the two.

With d, l and r_1, r_2, and r_3, as given above, define:

Q capacity of the queue,
λ total arrival rate,
s set-up time, and
p processing time.

A schematic representation of the system is given in figure 1.

3.2. *Alternative rules*

The alternative scheduling rules correspond to different queue disciplines in the context of queuing theory. The alternatives considered for the selection of the next job from the queue are as follows.

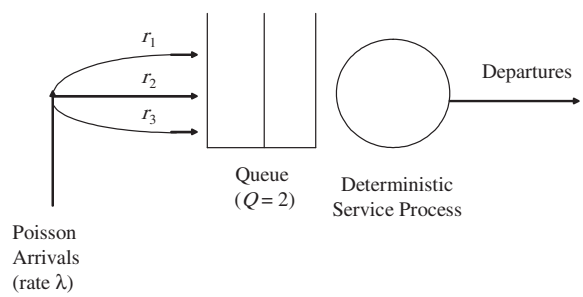

Figure 1. A schematic representation of the queuing system.

1. Earliest Due Date (EDD)
   - Process the first job in the queue. Note that since the time from the arrival of a job until its due date is a pre-specified constant, d, the first job in the queue has the earliest due date.
2. Shortest Set-up Time (SST)
   - If there is nothing in the queue maintain the latest set-up,
   - otherwise if there is at least one job from the family of the current set-up in the queue, process the next job from that family,
   - otherwise, process the jobs in rotating family index order starting from the index of the current set-up. If there are multiple jobs from the selected family, process the first one in the queue.
3. Priority Grouping (PG)
   - Form priority groups according to due dates such that if the time until the due date for a job is less than or equal to a pre-specified value, that job is classified as of 'high priority'. Otherwise it is classified as of 'low priority'. Then use the SST rule within the priority groups.

3.3. *Model formulation*

The problem can be modelled and solved as a finite state, undiscounted, discrete-time Markov Decision Process (MDP). A dynamic programming based methodology for MDPs was proposed by Howard (1960). White (1963) developed a successive approximation technique to solve the systems of equations arising in the context of undiscounted, discrete time MDPs. We use a variant of White's procedure.

The state of the system is defined as $[a(1), b(1), \ldots, a(Q+1), b(Q+1), c, ss]$ where

$a(i)$ = Family of the job indexed i ($a(i) = 0,1,2,3$ where $a(i) = 0$ indicates that there is no job indexed at i),
$b(i)$ = Time until the due date of the job indexed i ($b(i) = -l, -l+1, \ldots, 0,1,2, \ldots, d$),
c = Index number of the job in process, if the system is not empty; or the current set-up, if the system is empty,
ss = Stage of the service in progress ($ss = 0,1, \ldots, s, s+1, \ldots, s+p$).

The alternative scheduling rules constitute the alternatives of the MDP. A constant cost/period is incurred for carrying each item that is late by at least l time units.

3.4. *Generalization of the optimal control policy*

A generalization of the optimal scheduling policy from the MDP based on the experimental results is presented below. It is verified by inserting the generalized policy back into the MDP and comparing the cost to that of the optimal policy. Experimental results show that the performance is very close to optimal (Taner 2001).

The generalized policy requires scaling factors τ and γ, where τ represents a lower limit on the length of time that a job should have until its due date for it to be considered not urgent and γ represents an upper limit on the length of time that a job should have until its due date to be considered extremely tardy.

Generalized policy

- If there is no job in the system then use SST.
- Else if there is only one job in the system, then
  - If the job in the queue has τ time units or more until its due date and it has a bigger family index than the current set-up, use SST,
  - otherwise, use EDD.
- Else if there is more than one job in the queue
  - If all jobs waiting in the queue are already tardy and at least one is tardy γ or more time units, use the PG rule,
  - else if SST prefers a job in the queue that has τ time units or more until its due date while a more urgent job is also waiting for service, use EDD,
  - else if SST and EDD prefer the same job in the queue for subsequent processing use EDD,
  - else use SST.

The purpose of this policy is to facilitate effective utilization of the production resources and to meet the operation due dates. When a job finishes processing, selection of the next job to be processed is made so as to minimize set-ups. However, jobs with tight operation due dates that require set-ups, are also accounted for. A job that would potentially yield the maximum lateness can be given higher priority than other jobs even if it requires a set-up and the others do not.

4. Virtual factory job shop model

The Virtual Factory is a deterministic, iterative/adaptive simulation model developed by Hodgson *et al.* (1998, 2000) to solve standard job shop problems. The Virtual Factory models both single and batch processors, and yields very good solutions for industrial-sized problems (Hodgson *et al.* 1998, Thoney 2000 and Weintraub *et al.* 1999). The scheduling procedure in the Virtual Factory is based on an approach that was first proposed by Lawrence and Morton (1986), and Vepsalainen and Morton (1988). The approach consists of repeatedly simulating the system to be scheduled while simultaneously updating job sequences based on the results of the previous simulation. During the first iteration, jobs are sequenced on machines in order of increasing slack. Let d_i be the due date of job i and p_{ij} be the processing time of job i on machine j. The slack of job i on machine m is computed as

$$slack_{im} = d_i - \sum_{j \in m+} p_{ij},$$

where $m+$ is the set of all operations on job i's route subsequent to the one performed on machine m. Slack is a measure of the amount of time a job can queue and still meet its deadline. In general, however, dispatching in order of increasing slack may not provide good results, Caroll (1965). This can be attributed to the fact that slack does not take queuing time into account. In subsequent iterations, the queuing time of the previous iteration is used to modify slack, and jobs are sequenced in order of revised slack. Let q_{ij} be the queuing time of job i at machine j. Revised slack is computed as

$$slack'_{im} = d_i - \sum_{j \in m+} p_{ij} - \sum_{j \in m+^*} q_{ij},$$

where $m+*$ is the set of all operations on job i's route subsequent to the one performed on machine m except the one immediately following m. The slack calculation introduces an intermediate due date for the job. It is the time at which the job needs to begin processing at the next machine to meet its overall due date. This definition was found to be more effective in the procedure than a slack that includes p_{im} and q_{im}.

The procedure is run for a fixed number of iterations and the best solution is saved. The procedure tends to converge monotonically over the first 10 iterations or so (regardless of the size of the problem). In other words, the queuing time estimates become more accurate after the first iterations, producing better and better solutions. After that, the solutions tend to 'bounce' with no particular pattern involved. However, additional iterations typically yield better solutions.

5. Scheduling with set-ups and the job shop model

Job shop scenarios are modelled running the Virtual Factory for 200 iterations. Incorporation of set-up considerations into the Virtual Factory requires some additions and modifications. The set-up structure is assumed to be the same as discussed earlier. That is, processing a job with a smaller family index following one with a larger family index requires a set-up of s time units. No set-up is required if a job with the same or a larger family index is to be processed following the current job on the machine.

Revised slack computations are further modified to include sequence dependent set-up considerations. Define:

s_{ij} the set-up time on machine j to process job i, $s_{ij} \in \{0,s\}$ and
s_{ij}^{+} the part of s_{ij} that is incurred after the arrival of job j at machine i.

Modified revised slack is computed as

$$slack''_{im} = d_i - \sum_{j \in m+} p_{ij} - \sum_{j \in m+} s_{ij}^{+} - \sum_{j \in m+^*} q_{ij},$$

for each job i on every machine m.

The results gathered from the MDP model suggest that the current way revised slack is being used in the Virtual Factory may be inappropriate for machines involving sequence dependent set-up times. Sequencing jobs in revised slack order on such machines can be considered equivalent to using the EDD rule for local dispatching decisions. However, results with single machine problems indicate that the EDD rule may not be optimal in all cases. Hence, the generalized scheduling policy developed based on the experimental results with the Markov Decision Model, is implemented in the Virtual Factory.

Following Ovacik and Uzsoy (1994a), a 'forecasting' mechanism is incorporated into the heuristic policy to utilize global job shop status information for local scheduling decisions at the machine level. The jobs that have their next operation on the machine and that are estimated to arrive within a given 'forecast' horizon, T, as well as those waiting in the queue are considered as candidates for the current scheduling decision. The set J contains the candidate jobs. In some cases, the selected job may not be available for processing on the machine at the time the scheduling decision is made. In those cases, if there is some other job that can be completed before the estimated arrival of the selected job and processing it does not delay the completion of the set-up for the selected job past its estimated arrival time, then it is

processed next. If no such job exists in the queue, the set-up for the selected job is initiated. If set-up is completed before the arrival of the selected job, the machine is kept idle until its arrival.

5.1. *Heuristic policies based on the MDP*

Recall that the generalized policy requires two scaling factors, τ and γ, be set by the user. These factors are initialized in the first iteration based on the findings of some trial experimentation and they are updated in each successive iteration using the results of the previous iteration. Two versions of the heuristic policy, which differ in the way these two factors are used, are presented below.

Scheme 1: Policy – Version 1

In this version of the heuristic policy, the length of time from the point that the machine becomes available until the operation due date ($slack''$) of a candidate job is used to determine if that job should be considered 'not urgent' or 'extremely tardy'. In particular, a job is considered not urgent if it has at least τ time units from the time that the scheduling decision is made until its operation due date ($slack''$). Likewise, a job is considered extremely tardy if it has γ time units or less from the time that the scheduling decision is made until its operation due date ($slack''$).

Scheme 2: Policy – Version 2

This version of the policy is essentially the same for those candidate jobs that are already in the queue at the time the machine becomes available. However, when a candidate job is estimated to arrive after the machine becomes available, the length of time from the estimated arrival time of the candidate job until its operation due date ($slack''$) is used to determine if that job should be considered not urgent or extremely tardy. In particular, a job is considered not urgent if it has at least τ time units from its estimated arrival time until its operation due date ($slack''$). Likewise, a job is considered extremely tardy if it has γ time units or less from its estimated arrival time until its operation due date ($slack''$).

5.2. *Heuristic policies from the literature*

The computational intractability of job shop scheduling problems rules out the possibility of obtaining optimal solutions for the scenarios to be analyzed. Thus, the results need to be compared to a lower bound or to the results of previously proposed algorithms. However, developing a lower bound for the problem under consideration is equivalent to establishing a lower bound for the asymmetric travelling salesman problem with time windows. In spite of the special structure of the set-up matrix at hand, developing a tight lower bound that can be computed in a reasonable amount of time is extremely difficult. Therefore, the performance of the technique is evaluated in relation to that of existing techniques.

To the best of the authors' knowledge, Ovacik and Uzsoy (1994a) is the only paper in the literature that addresses the dynamic job shop scheduling problem with sequence dependent set-up times and the objective of minimizing $L_{\max}$. Their technique extends four dispatching rules over a forecast window, T, and uses partial enumeration to determine the next job for processing each time a machine becomes available. They compare the performance of these four scheduling rules with that of the 'job earliest due date (J-EDD)', 'operation earliest due date (O-EDD)' and 'apparent tardiness cost with set-ups (ATCS)' dispatching rules. J-EDD and

O-EDD rules are currently used in industry for due-date related performance criteria. An explanation of the ATCS rule can be found in Bhaskaran and Pinedo (1991). The objective of this rule is to minimize the total weighted tardiness in a job shop with sequence dependent set-ups. Ovacik and Uzsoy (1994a) report that their algorithm LAO(β) is the best performer of the seven in terms of average solution quality and robustness. Hence, the LAO(β) rule is used as a benchmark in this study. The LAO(β) rule is explained and a modification of it is proposed below.

Scheme 3: LAO(β) Rule

Whenever a machine becomes available, a set, J, of candidate jobs is determined. First, β jobs with the earliest operation due dates in set J are sequenced optimally via enumeration, and the first job in the optimal sequence is selected. If the selected job is already in the queue, it is scheduled next. Otherwise, the jobs in the queue are considered in increasing earliest operation due date order, and the first job that can be finished by the estimated arrival time of the selected job is scheduled next. If no such job exists in the queue, the machine is kept idle until the arrival of the selected job. This technique is more robust than Schemes 1 and 2 in the sense that it does not require any special set-up structure to produce good schedules.

Scheme 4: Modified LAO(β) Rule

In many industrial applications, the set-up operations are separable in the sense that changeover for a certain job can start before the arrival of that job at the machine. For example, in the context of textile dyeing processes, the dyeing vessel can be cleaned for dyeing a lighter coloured fabric before that fabric actually arrives at the machine. Thus, the original $LAO(\beta)$ Rule is modified to allow starting the set-up for a selected job before its arrival at the machine. When a selected job is not already in the queue, jobs in the queue are considered in earliest operation due date order, and the first one that can be finished without delaying the completion of set-up for the selected job past its estimated arrival time is scheduled next. If no such job exists in the queue, the set-up for the selected job is initiated. If the set-up is completed before the selected job arrives, the machine is kept idle until the job becomes available.

Ovacik and Uzsoy (1994a) estimate the operation due date of job i at machine j by subtracting from its due date a multiple, k, of its total downstream processing and average set-up times. In the implementation of this study, the modified revised slack ($slack_{im}''$) values for each job i on every machine m are used as operation due dates in Schemes 1 through 4.

6. Experimental framework

Two general job shop settings are analyzed. The first setting is similar to the classical job shop setting studied by Ovacik and Uzsoy (1994a), where all machines have sequence dependent set-ups and each job visits every machine exactly once in some random order. In the second setting, only one of the machines has sequence dependent set-ups and all jobs go through this machine at the midpoint of their otherwise randomly generated operation routes.

6.1. *Sequence dependent set-ups on all machines*

Scenarios with all combinations of 20, 50 and 100 jobs, and 5, 11 and 21 machines are generated. Processing times are sampled from a Uniform distribution

Number of jobs	20	50	100
Number of machines	5	11	21
Set-up time	66	200	600
Due date range	Low	Medium	High

Table 1. Experimental design.

with a range of [1,200]. The non-zero set-up time value s is varied as 66, 200 and 600. These numbers were arbitrarily chosen as one third of, the same as, and threefold of the maximum processing time of 200 in order to generate problems in which the set-up times are smaller than, in the same range as, and larger than the processing times, respectively. Due dates are sampled from a Uniform distribution between 1 and $1 + D$, where D is the due date range. The due date range is varied based on Γ defined as follows:

$$\Gamma = M(100.5 + 0.5s),$$

where M is the number of machines. Problems with low, medium and high due date ranges are generated by setting D equal to $0.5 \times \Gamma$, Γ and $2 \times \Gamma$, respectively. Twenty random instances (replications) are generated for each of the 81 parameter combinations shown in table 1, resulting in a total of 1620 problems.

6.2. *Sequence dependent set-ups on a single machine*

Scenarios with all combinations of 20, 50 and 100 jobs, and 5, 11 and 21 machines are generated. The number of operations is coordinated with the number of machines. Processing times on regular machines are sampled from a Uniform distribution with a range of [1,200]. Processing and set-up times on the machine with sequence dependent set-ups are generated such that the machine will be either be *a bottleneck* (*B*), *somewhat of a bottleneck* (*SWB*) or *not a bottleneck* (*NB*). Define

$$f = \frac{P}{P + 0.5s}, \tag{1}$$

where $P + 0.5$ is the average processing time and s is the set-up time on the machine with sequence dependencies. The value of $P + 0.5s$ is determined as follows:

$$k(P + 0.5s) = \frac{O - 1}{M - 1} \times 100.5, \tag{2}$$

where k is a factor that determines the load on the machine with sequence dependencies, O is the number of operations and M is the number of machines. The average processing time on a regular machine is 100.5. Factor k is set equal to 0.5, 1 and 2 to make the machine with sequence dependencies *a bottleneck*, *somewhat of a bottleneck* and *not a bottleneck*, respectively. Factor f is arbitrarily varied as 0.25, 0.50 and 0.75 to generate different P/s ratios. This makes the ratio of the set-up time (s) to the maximum processing time ($2P$) 1/3, 1 and 3, respectively. Individual values of P and s are determined by solving equations (1) and (2) simultaneously. The processing time on the machine with sequence dependent set-ups is sampled from a Uniform distribution with a range of [1,2P]. Due dates are sampled from a Uniform distribution with a range between [1,1 + D], where D is the due date

Number of jobs	20	50	100
Number of machines/operations, M/O	5/5	11/7	21/11
Bottleneck status	Yes	Somewhat	No
F	0.25	0.50	0.75
Due date range (D)	Low	Medium	High

Table 2. Experimental design.

range parameter. Due date range parameter D is varied based on Ω defined as follows:

$$\Omega = (O - 1) \times 100.5 + P + 0.5s.$$

Problems with low, medium and high due date ranges are generated by setting D equal to $0.5 \times \Omega$, Ω, and $2 \times \Omega$, respectively. Twenty random instances (replications) are generated for each of the 243 parameter combinations shown in table 2 resulting in a total of 4860 problems.

7. Computational results

Since the experimental results depend upon the way T, β, γ and τ are selected, these parameters are determined in a common manner in all experiments. This makes an accurate comparison of the scheduling schemes possible. T is set equal to 150, and β is chosen as 3, based on the results of some initial experimentation. Parameters γ and τ are initialized as 0 and 150, respectively, at the first iteration, and updated at each iteration by subtracting from these initial values the $L_{\max}$ obtained during the previous iteration.

Following Ovacik and Uzsoy (1994a), the approach described by Lenstra (1977) is used to compare the different scheduling methods used in this study. For each scheme, the average of

$$G = \frac{L_{\max} + d_{\max}}{\text{BEST} + d_{\max}}$$

over all instances of a given problem is reported, where $L_{\max}$ is the value obtained by a given method, $d_{\max}$ is the maximum due date for a particular instance of the problem, and BEST is the smallest $L_{\max}$ value found for that instance by any of the four methods studied. The constant, $d_{\max}$ is added to both the numerator and the denominator of the ratio in order to avoid problems caused by non-positive $L_{\max}$ values. Clearly, this ratio should be close to 1.0 for a successful method.

When analyzing the results, it is important to remember that the $LAO(\beta)$ rule (scheme 3) and the modified $LAO(\beta)$ rule (scheme 4) adapted from Ovacik and Uzsoy (1994a) do not require any special set-up structures, while either version of the policy (schemes 1 and 2) proposed in this study may not be effective for a different set-up structure.

7.1. *Sequence dependent set-ups on all machines*

Figure 2 shows the overall relative performance and computational time on a 400 MHz PC with scheduling schemes 1 through 4. Schemes 1 and 2 perform significantly better and run only a fraction of a second slower than schemes 3 and 4 on average. Performance of scheme 2 is slightly better than that of scheme 1.

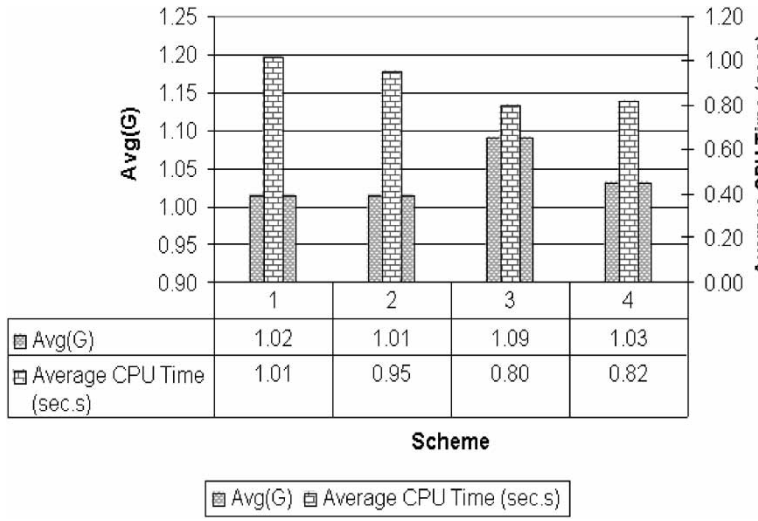

Figure 2. Relative performance and computational time of schemes 1–4.

Confidence level	$U1/2 < L3$	$L1/2 > U3$
95%	67 (83%)	0 (0%)
90%	71 (88%)	0 (0%)
75%	73 (90%)	0 (0%)
60%	76 (94%)	1 (1%)

Table 3. Statistical analysis of results when all machines have sequence dependent set-ups.

Ovacik and Uzsoy's (1994a) original scheme (scheme 3) is significantly improved by the proposed modifications (scheme 4) at a very modest increase in computational time. However, even this modified scheme seems inferior to either version (schemes 1 and 2) of the scheduling policy proposed in this study.

Lower and upper confidence limits are established on the mean of 20 replications for each one of the 81 combinations of the parameters systematically varied in the experimentation (table 1). Confidence levels ($100(1-\alpha)\%$) of 95%, 90%, 75% and 60% are used to determine the t-value. The upper confidence limit on the average $L_{\max}$ obtained from the better of schemes 1 and 2 ($U1/2$) is compared with the lower confidence limit on the average $L_{\max}$ obtained from scheme 3 ($L3$). When $U1/2$ is smaller than $L3$, the performance of the proposed scheduling technique is said to be statistically significantly better than that of scheme 3. In addition, the lower confidence limit on the average $L_{\max}$ obtained from the better of schemes 1 and 2 ($L1/2$) is compared with the upper confidence limit on the average $L_{\max}$ obtained from scheme 3 ($U3$). When $L1/2$ is greater than $U3$, the performance of the proposed technique is said to be statistically significantly worse than that of scheme 3. Table 3 summarizes the analysis. At a confidence level of 95%, the proposed technique statistically outperforms scheme 3 in 67 of the 81 cases (83%). This value increases for decreasing levels of confidence. The proposed technique performs statistically worse than scheme 3 in none of the cases with confidence levels greater than 75%

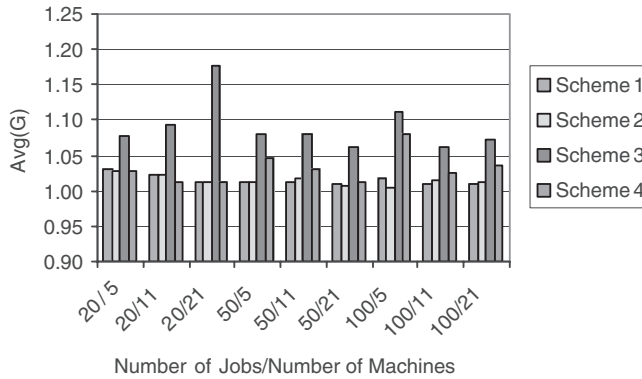

Figure 3. Effect of varying the number of jobs and the number of machines on the relative performance of schemes 1–4.

and in only one of the 81 cases when the confidence level is 60%. Although the confidence levels smaller than 90% generally do not provide strong statistical interpretations, results at confidence levels of 75% and 60% are included to give the reader a sense for the practical superiority of the proposed schemes.

In the following paragraphs, a p-value is provided for the comparisons made on the mean performance of different scheduling schemes. These p-values indicate the minimum level of significance that leads to rejection of the statistical hypothesis that the means of the particular schemes are equal under the stated cases.

Figure 3 shows the effect of varying the number of jobs and the number of machines on the relative performance of schemes 1 through 4. Scheme 3 is the worst performer in all cases. When the number of jobs is 20, performances of schemes 1, 2 and 4 are close to each other. When the number of jobs is 50 or 100, schemes 1 and 2 perform almost equally well. Performance of scheme 4 is worse than that of schemes 1 and 2 (p-value = 0.000 in both cases). Scheme 2 outperforms scheme 1 when there are 50 jobs and 21 machines, and 100 jobs and 5 machines (p-value = 0.157). Scheme 1 is the best performer in all other cases closely followed by scheme 2 (p-value = 0.599). When the number of machines is fixed at 5 and the number of jobs is increased, the difference in performance of schemes 1 and 2, and schemes 3 and 4 becomes more apparent. The impact of varying the number of jobs in the same way seems less significant when there are 11 or 21 machines in the shop.

Figure 4 shows the effect of varying the number of jobs and the due date range on the relative performance of schemes 1 through 4. When there are 20 jobs, performances of schemes 1, 2 and 4 are close to each other, and significantly better than that of scheme 3 under any due date range (p-values of 0.037, 0.072 and 0.000, respectively for schemes 1, 2 and 4 against scheme 3). Scheme 1 is the best performer with low and medium due date ranges (p-value = 0.000), and scheme 4 is the best performer with a high due date range (p-value = 0.000). The effectiveness of scheme 4 with a high due date range may not be surprising, as one should expect this scheme explicitly to consider virtually all schedules that may yield the best solution when there is little or no congestion at key resources. When the number of jobs is 50 or 100, schemes 1 and 2 perform almost equally well; however, the performance of schemes 3 and 4 are significantly worse in most cases (p-value = 0.000). Scheme 4 performs slightly better than schemes 1 and 2 when there are 50 jobs and the due date

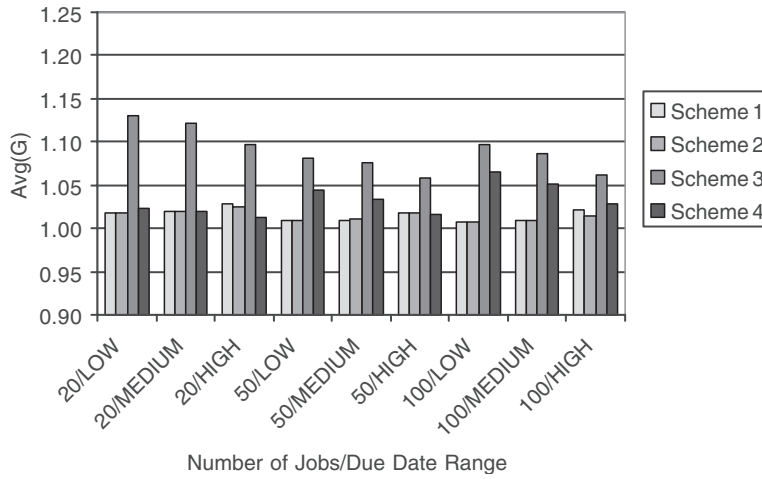

Figure 4. Effect of varying the number of jobs and due date range on the relative performance of schemes 1–4.

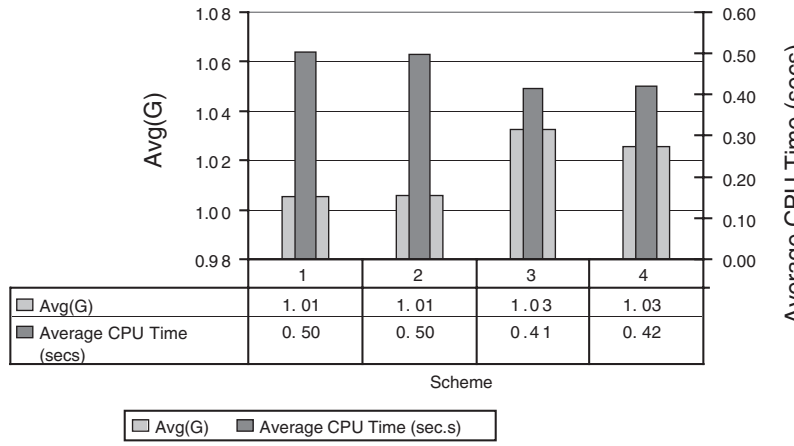

Figure 5. Average relative performance and CPU time of schemes 1–4.

range is high (p-value = 0.000 against both schemes 1 and 2). Scheme 3 is the worst performer in all cases. For a given number of jobs, the difference in the performance of schemes 1 and 2, and schemes 3 and 4 gets less significant for increasing due date ranges. This may suggest that the methods proposed in this study are more effective in comparison to the benchmark techniques when there are many jobs competing for the same machines.

7.2. *Sequence dependent set-ups on a single machine*

Figure 5 shows the relative performance and computational time with scheduling schemes 1 through 4 on average. Schemes 1 and 2 perform almost equally well. The average performance with schemes 3 and 4 is worse than that with schemes 1 and 2. Average computational times on a 400 MHz PC are less than 1 second with all

schemes. This suggests that the modifications proposed in this study to improve the original technique developed by Ovacik and Uzsoy (1994a) are useful.

Lower and upper confidence limits are established on the mean of 20 replications for each one of the 243 combinations of the parameters systematically varied in the experimentation (table 2). Table 4 summarizes the analysis. At a confidence level of 95%, the proposed technique statistically outperforms scheme 3 in 56 of the 243 cases (23%). This value increases for decreasing levels of confidence, reaching 190 (79%) with a confidence level of 60%. The proposed technique performs statistically worse than scheme 3 in none of the cases with confidence levels greater than 75% and in only one of the 243 cases when the confidence level is 60%. Performances of schemes 1 and 2 and that of scheme 3 are closer when only one machine has sequence dependent set-ups. This may be because performance depends more on scheduling of the sequence independent (regular) machines.

As in the previous section, *p*-values are provided for the statements made on the mean performance of different schemes under specific cases in the following paragraphs.

Figure 6 shows the effect of varying the number of jobs, machines and operations on the relative average performance of scheduling schemes 1 through 4. In general, schemes 1 and 2 produce significantly better schedules than schemes 3 and 4 (p-value = 0.000). Modifications proposed in this study on the original technique developed by Ovacik and Uzsoy (1994a) seem to improve the average performance, as the results are better with scheme 4 than scheme 3 (p-value = 0.000). Differences in performance among the scheduling schemes become more significant when the number of jobs is increased for a given number of machines and operations.

Confidence level (%)	$U1/2 < L3$	$L1/2 > U3$
95	56 (23%)	0 (0%)
90	80 (30%)	0 (0%)
75	119 (50%)	0 (0%)
60	190 (79%)	1 (0%)

Table 4. Statistical analysis of results when only one machine has sequence dependent set-ups.

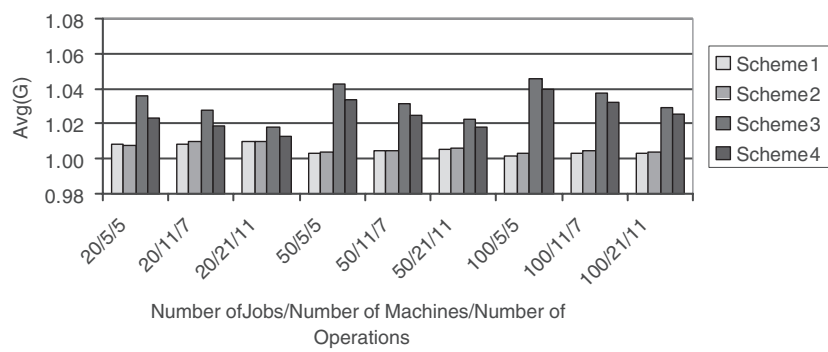

Figure 6. Effect of varying the number of jobs, machines and operations on the relative performance of schemes 1–4.

Figure 7 shows the effect of varying the level of congestion at the machine with sequence dependent set-ups for 20, 50 and 100 job problems. Scheme 1 is the best method in all cases except when there are 20 or 100 machines and the machine with sequence dependent set-up times is not a bottleneck (p-value = 0.000). Scheme 3 is the worst method in all cases. Differences in performance of different scheduling schemes for any given number of jobs are more significant in the cases when the machine with sequence dependent set-ups is more of a bottleneck.

Figure 8 shows the effect of varying the number of jobs and the due date range on the relative performance of scheduling schemes 1 through 4. Scheme 1 and scheme 3, respectively, are the best and worst performing scheduling methods in all cases. The difference in average performance of the four schemes becomes less significant as the due date range is increased for a given number of jobs, and as the number of jobs is decreased for a given due date range. Hence, bigger differences in performance are observed when there are many jobs competing for the same machines and when the

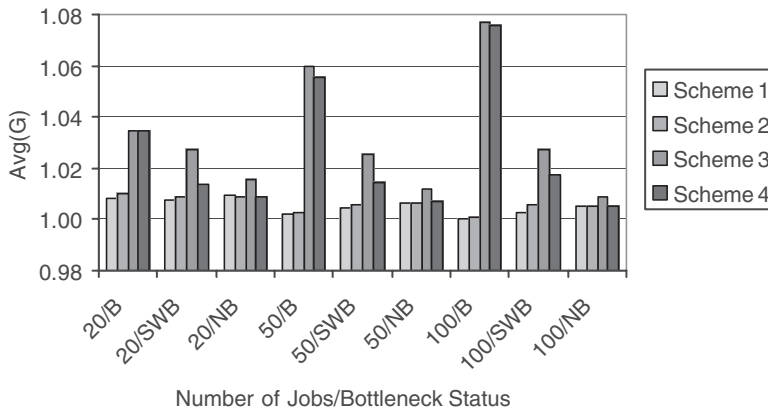

Figure 7. Effect of varying the number of jobs and the level of congestion at the machine with sequence dependencies on the relative performance of schemes 1–4.

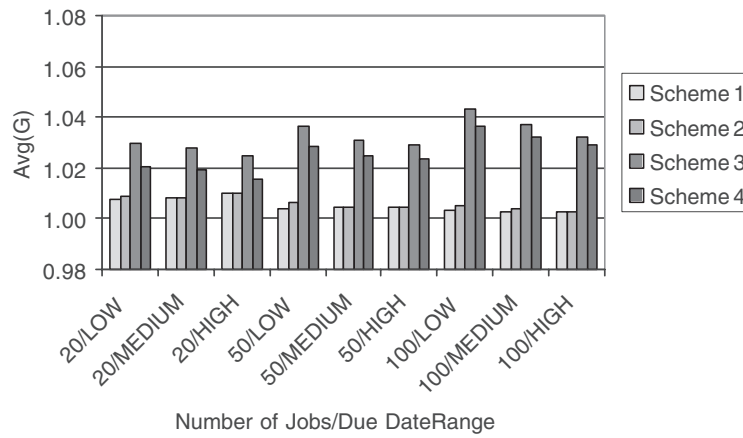

Figure 8. Effect of varying the number of jobs and due date range on the relative performance of schemes 1–4.

number of alternative schedules that may yield good results is small. Therefore, the methods proposed in this study are more effective in solving difficult problems than the benchmark methods.

8. Conclusion and future research

Two general job shop settings are analyzed. In either setting the proposed schemes 1 and 2 outperform benchmark schemes 3 and 4 on average. In general, the performances of schemes 1 and 2 are almost equally good. Scheme 4, which is a modification of the LAO(β), rule has a better average performance than that of the original LAO(β) rule (scheme 3). Finally, differences in performance of the four scheduling schemes become more significant when more jobs compete for the same key resources in the shop.

The basic set-up structure examined is one of the simplest of those seen in practice. It is necessary to address scenarios with more general set-up structures and explore scheduling methodologies in the light of the findings of this study. Bounding strategies for the problem need to be investigated. Obtaining a good theoretical lower bound seems extremely difficult, but it may be possible to construct statistical lower bounds at least for some key scenarios (see for example Golden and Alt 1979). Modelling and solving parallel machine problems using the Virtual Factory is another fertile but challenging area of research.

References

Allahverdi, A., Gupta, J. N. D. and Aldowaisan, T., 1999, A review of scheduling research involving set-up considerations. *OMEGA, International Journal of Management Science*, **27**, 219–239.

Bhaskaran, K. and Pinedo, M., 1991, Dispatching. In G. Salvendy (ed.), *Handbook of Industrial Engineering* (New York: Wiley), chapter 83.

Caroll, D. C., 1965, Heuristic sequencing of single and multiple component jobs. Unpublished Ph.D. Dissertation, MIT, USA.

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

Golden, B. L. and Alt, F. B., 1979, Interval estimation of a global optimum for large combinatorial problems. *Naval Research Logistics Quarterly*, **26**(1), 69–77.

Hodgson, T. J., King, R. E., Monteith, S. K. and Schultz, S. R., 1987, Developing control rules for an AGVS using Markov decision processes. *Material Flow*, **4**(1), 87–98.

Hodgson, T. J., Cormier, D., Weintraub, A. J. and Zozom, A., 1998, Satisfying due-dates in large job shops. *Management Science*, **44**, 1442–1446.

Hodgson, T. J., King, R. E., Thoney, K. A., Stanislaw, N., Weintraub, A. J. and Zozom, A., 2000, On satisfying due-dates in large job shops: idle time insertion. *IIE Transactions*, **32**, 177–180.

Howard, R. A., 1960, *Dynamic Programming and Markov Processes* (New York: MIT Press).

Lawrence, S. R. and Morton, T. E., 1986, Patriarch: hierarchical production scheduling. *National Bureau of Standards Special Publication*, **724**, 87–97.

Lenstra, J. K., 1977, Sequencing by enumerative methods. Mathematical Center Tract 69, Amsterdam: Mathematisch Centrum.

Matsui, M. and Shinghu, T., 1978, A queueing analysis of conveyor-serviced production station and the optimal range strategy. *AIIE Transactions*, **10**(1), 89–99.

Monma, C. L. and Potts, C. N., 1989, On the complexity of scheduling with batch set-ups. *Operations Research*, **37**, 798–804.

Ovacik, I. M. and Uzsoy, R., 1994a, Exploiting shop floor status information to schedule complex job shops. *Journal of Manufacturing Systems*, **13**, 73–84.

Ovacik, I. M. and Uzsoy, R., 1994b, Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence dependent set-up times. *International Journal of Production Research*, **32**, 1243–1263.

PINEDO, M., 1995, Open shop and job shop scheduling. In *Scheduling: Theory*, *Algorithms*, *and Systems* (New Jersey: Prentice Hall), pp. 118–141.

TANER, M. R., 2001, Scheduling with sequence dependent set-up times. Unpublished Ph.D. Dissertation, North Carolina State University, Raleigh, NC, USA.

THONEY, K. A., 2000, Simultaneous plant and supply chain scheduling. Unpublished Ph.D. Dissertation, North Carolina State University, Raleigh, NC, USA.

UZSOY, R., LEE, C. Y. and MARTIN-VEGA, L. A., 1992, Scheduling semiconductor test operations: minimizing maximum lateness and number of tardy jobs on a single machine. *Naval Research Logistics*, **39**, 369–388.

UZSOY, R., MARTIN-VEGA, L. A., LEE, C. Y. and LEONARD, P. A., 1991, Production scheduling algorithms for a semiconductor test facility. *IEEE Transactions Semiconductor Manufacturing*, **4**, 270–280.

VEPSALAINEN, A. P. J. and MORTON, T. E., 1988, Improving local priority rules with global lead-time estimates: a simulation study. *Journal of Manufacturing and Operations Management*, **1**, 102–118.

WEINTRAUB, A., CORMIER, D., HODGSON, T. J., KING, R. E., WILSON, J. R. and ZOZOM, A., 1999, Scheduling with alternatives: a link between process planning and scheduling. *IIE Transactions*, **31**, 1093–1102.

WHITE, D. J., 1963, Dynamic programming, Markov chains and the method of successive approximations. *Journal of Mathematics Analysis and Applications*, **6**, 373–376.