

An analysis of heuristics in a dynamic job shop with weighted tardiness objectives

E. KUTANOGLU[†] and I. SABUNCUOGLU^{‡*}

Meeting due dates as a reflection of customer satisfaction is one of the scheduling criteria that is frequently encountered in today's manufacturing environments. The natural quantification of this qualitative goal involves tardiness related measures. In this study, we consider the dynamic job shop scheduling problem with the weighted tardiness criterion. After we present a comprehensive literature survey on the topic, we measure the long-run performances of more than 20 single-pass dispatching rules under various experimental conditions. In this study, we pay special attention to recently proposed dispatching heuristics such as CEXSPT, CR+ SPT, S/RPT+ SPT, and Bottleneck Dynamics (BD). We also investigate the effects of six resource pricing schemes proposed recently for BD. Moreover, we extend the earlier versions of inserted idleness and identify the conditions in which these techniques can be applied without incurring too much computational cost. Future research directions are also outlined in light of the computational results.

1. Introduction

Even though flexible manufacturing systems and computer integrated manufacturing are today's key words that frequently appear in many research agendas, scheduling of job shops still receives ample attention from both researchers and practitioners. A part of the reason is that job shop problems in various forms still exist in most of the advanced manufacturing systems. Besides, analysis of job shop scheduling problems provides important insights into the solution of the scheduling problems encountered in more realistic and complicated systems.

Meeting due dates, as a reflection of customer satisfaction, is one of the scheduling criteria that is frequently encountered in practical problems. The natural quantification of this qualitative goal involves the tardiness measure. Tardiness is the positive lateness a job incurs if it is completed after its due date. When the jobs have different importance levels or weights, the resulting problem is called a weighted tardiness problem. In general, weighted tardiness problems (even the single machine version) are NP-hard (Lenstra and Rinnooy Kan 1978). Hence, heuristics are generally preferred especially for dynamic and stochastic job shop problems. (In this study, we assume that all the jobs are accepted for processing and each job has an externally set due date.)

These heuristics can be classified into two categories: (1) *single-pass* (one-pass) and (2) *multi-pass* heuristics. In one-pass heuristics, a single complete solution is built up one step at a time. Most priority dispatching rules proposed in the literature

Revision received February 1998.

[†] Department of Industrial and Manufacturing Systems Engineering, Bethlehem, PA 18015, USA. This research was done while E. Kutanoğlu was at Bilkent University.

[‡] Bilkent University, Department of Industrial Engineering, Bilkent, Ankara, 06533 Turkey.

* To whom correspondence should be addressed.

(Panwalker and Iskander 1977) can be considered in this category. In multi-pass heuristics, an initial schedule is generated in the first pass, and then the consecutive passes search, for improvement in performance. In this category, we can list neighbourhood search, tabu search, simulated annealing, and iterative dispatching (see Morton and Pentico (1993) for further discussion).

We focus on the single-pass heuristics, specifically priority dispatch rules. As we discuss in the next section, there are studies which investigated performances of dispatching rules by using discrete-event simulation. The general picture seems to be that no single rule is the best under all possible conditions and the relative performances of the rules are affected by factors such as shop load level, due date tightness, and scheduling criteria. Since all these studies have been done in different operational settings, the results sometimes give conflicting reports. Hence, there is a need for comprehensive research and testing to clarify these points, which is one objective of this study. The rules evaluated in this study are listed in Appendix B along with the notation as shown in Appendix A.

Some of these rules have been recently developed (such as CEXSPT, MDSPRO, CR+ SPT, S/RPT+ SPT, and BD) and hence their performances are not generally known in the literature. One goal of this study is to compare these new rules with the existing ones so that a more complete picture can be drawn. Note that the majority of these new rules are more complex and they demand more job and shop-related information, which may not be available in unsophisticated systems. In this paper, we will also show whether these complex heuristics can justify their high information need with higher performance quality.

We also observed that the majority of the existing studies have been done in a uniform (or balanced) environment (i.e. all machines are approximately equally utilized). Their relative performances are not generally known for the unbalanced systems (i.e. non-uniform or bottleneck case). In this paper, we also analyse the bottleneck case to see if the conclusions drawn from the uniform systems are still valid.

We employ different resource pricing schemes for BD, most of which have been proposed in Morton and Pentico (1993). Some of these are tested for the first time in a dynamic job shop environment. Since the mechanisms involved in resource pricing distinguish loaded machines from less utilized ones, the performance of resource pricing is expected to be dependent on whether the shop is balanced or unbalanced.

It is well known that schedules without idle time constitute the dominant set of schedules in static single machine problems with regular performance measures and that the semi-active schedules constitute a dominant set for dynamic problems (Baker 1974). Unfortunately, dispatching heuristics normally generate a non-delay schedule which does not include inserted idleness. However, with little additional computation, some heuristics can be modified to myopically search possibilities with inserted idleness. One such method is proposed by Morton and Ramnath (1992) for a dynamic single machine problem. In this study, we extend this technique for BD and implement it for a job shop. By considering inserted idleness, we will try to identify the conditions under which it may be better to keep the resource idle for a soon-to-arrive urgent job.

In our experiments, we did not use multi-pass heuristics such as the lead time iteration (LTI) method. First of all, such an iterative approach assumes either a static environment or perfect information on the system's future in the dynamic case. Since we consider unknown job arrivals we cannot directly employ these

multi-pass mechanisms. More importantly, we think that LTI and other multi-pass heuristics are search heuristics. Since the aim of this study is to test *dispatching heuristics*, we only included the dispatch versions of BD. We think that search heuristics and dispatching rules are two different categories that should be compared with careful analysis of solution quality versus computational cost.

The rest of the paper is organized as follows: we present a comprehensive review of the relevant literature along with the discussion of the rules tested in section 2. Section 3 gives the system considerations and experimental conditions. Computational results are presented in section 4. (We primarily reported weighted tardiness, although we present selected results for percent tardy jobs and conditional weighted tardiness to initiate deeper discussion.) The paper ends with concluding remarks and suggestions for future research in section 5.

2. Review of literature

In this study, we only consider systems where jobs are independent. Hence, we did not include assembly type work in which coordination of jobs plays a more important role than dispatching. As stated earlier, there are a number of dispatching rules proposed in the job shop literature. Among them, some are very simple, known and used for many years. We first review these conventional rules. Then we discuss more recent studies especially on bottleneck dynamics. Table 1 lists these papers along with shop type, performance measure and rules tested.

2.1. Conventional priority dispatching rules

The information needed by dispatching rules are classified by Ramasesh (1990) according to their information content as follows:

- arrival times (e.g. FCFS),
- processing times (e.g. SPT),
- due date information,
 - allowance-based (e.g. EDD),
 - slack based (e.g. SLACK),
 - ratio-based (e.g. CR, S/RPT, S/OPN, A/OPN),
- combination of one or more of the above (e.g. WSPT, MOD, ODD, OSLACK).

Among the rules, FCFS is generally used as a benchmark. WSPT is the weighted version of the shortest processing time (SPT) rule. *Flow allowance* of a job is the time between the release date and the due date, $A_i = d_i - r_i$. The remaining allowance of a job i at time t is calculated as $A_i(t) = d_i - t$. Since t is the same for all jobs, the simplest version of the allowance based priority is the earliest due date rule (EDD). The *global slack time* $S_{ij}(t)$ of a job is time after remaining work is deducted from allowance ($S_{ij}(t) = A_i(t) - P_{ij}$ where P_{ij} is the total remaining processing time). The simplest slack-based priority rule is the slack time rule (SLACK), which gives priority to the job with the smallest $S_{ij}(t)$.

The ratio-based rules utilize a kind of ratio in their implementation. For instance, critical ratio rule (CR) gives priority to the job with the smallest $A_i(t)/P_{ij}$. Another ratio-based rule is slack per remaining processing time (S/RPT) with the priority index $S_{ij}(t)/P_{ij}$. The priority index of the slack per remaining operation rule (S/OPN) is calculated as $S_{ij}(t)/m_{ij}$, where m_{ij} is the remaining number of operations from operation j to the last operation. S/RPT gives priority to the job with the longer remaining processing time, while S/OPN considers the job with more operations

Authors	Conditions	Measure	Rules tested
Gere (1966)	Dynamic JS	Unweighted T	FCFS, SPT, SLACK SLACK/A , EDD, S/OPN
Elvers (1973)	Dynamic JS	Percent T	FCFS, EDD, LWKR, SPT, SLACK , S/OPN, CR
Weeks (1979)	Dynamic JS	Unweighted L	SPT, S/OPN
Miyazaki (1981)	Dynamic JS	Unweighted T	FCFS, S/OPN, A/OPN, SLACK/A
Elvers and Taube (1983)	Dynamic JS	Unweighted T	SPT, EDD, SLACK, S/RPT
Kanet and Hayya (1982)	Dynamic JS	Unweighted T	SPT, EDD, SLACK, CR, ODD , OSLACK, OCR
Baker and Kanet (1983)	Dynamic JS	Unweighted T	COVERT, CR, S/RPT, SLACK , MDD, MOD
Baker (1984)	Dynamic JS	Unweighted T	S/OPN, A/OPN, MDD, MOD
Carroll (1965)	Dynamic JS	Unweighted T	SPT, S/OPN, COVERT
Russell <i>et al.</i> (1987)	Dynamic JS	Unweighted T	EDD, SLACK, S/OPN, SPT, MDD, MOD, AU, COVERT
Shultz (1989)	Dynamic JS	Unweighted T	SPT, S/OPN, OCR, MOD, COVERT , CEXSPT
Anderson and Nyirenda (1990)	Dynamic JS	Weighted T	WSPT, MOD, COVERT, CEXSPT , CR+ SPT , S/RPT+ SPT
Vepsalainen and Morton (1987)	Dynamic JS	Weighted T	FCFS, WSPT, EDD, S/RPT, COVERT , ATC
Vepsalainen and Morton (1988)	Dynamic JS	Weighted T	FCFS, WSPT, EDD, S/OPN, COVERT , ATC
Kanet and Zhou (1993)	Dynamic JS	Unweighted T	FCFS, SPT, MOD, COVERT, ATC , MEANP
Ovacik and Uzsoy (1994)	Dynamic JS	Maximum L	EDD, ODD , ATC
Morton <i>et al.</i> (1988)	Static FS and JS	NPV	CR, COVERT, EXP-ET, SCHED-STAR
Lawrence and Morton (1993b)	Static RCPS	Weighted T	20 dispatching rules, pricing heuristics
Lawrence and Morton (1993a)	Static JS	Weighted T	FCFS, WSPT, ODD , OSLACK , ATC , BD

Table 1. Papers reviewed and their main characteristics (JS: Job Shop, FS: Flow Shop, RCPS: Resource-constrained project scheduling, T: Tardiness, L: Lateness, NPV: Net present value of revenues and costs). (The rule(s) found to be best in corresponding experimental study are in bold face.)

remaining as urgent. The S/RPT rule is equivalent to the one with CR in the sense that their priority indexes yield the same sequence (i.e. $S/RPT_{ij}(t) = CR_{ij}(t) - 1.0$). Another ratio rule is the slack/allowance ratio which gives priority to the job with minimum ratio of $SLACK/A_{ij}(t) = S_{ij}(t)/A_i(t)$.

Since a negative ratio is difficult to interpret, the ratio-based priority rules have drawbacks: when the remaining allowance or slack time is negative, these rules behave contrary to their intent. For example, the intent of the S/OPN rule is to give relatively higher priorities to the jobs which have more remaining operations because they may encounter more queuing delay. However, when the slack becomes negative it tends to misbehave as a result of assigning higher priorities to the jobs with fewer remaining operations. Kanet (1982) resolved this *anomaly* by proposing the rule called modified dynamic slack per remaining operation (MDSPRO).

All these rules except MDSPRO have been extensively tested in the previous studies although not every study included all of them and used the same test bed (i.e. experimental conditions). It seems that the relative performances of the rules are very sensitive to the system conditions. In general, EDD, SLACK, S/RPT perform well in light-loaded shop conditions, whereas SPT performs well in congested shops with tight due dates, but fails with light loads and loose due dates (Elvers 1983, Elvers and Taube 1983, Gere 1966, Miyazaki 1981, Weeks 1979).

Some rules utilize operation due dates. Among several ways of assigning operation due dates, the *work content method* is generally suggested for mean tardiness (Baker 1984). According to this method, the initial flow allowance of a job is allocated to the operations proportional to their processing times. The rules such as EDD, SLACK and CR have their operation due date versions (ODD, OSLACK and OCR). It seems that operation-based rules perform better than their job-based counterparts (Kanet and Hayya 1982).

There are also rules which combine the processing time and due date information. One such rule is the modified due date (MDD) in which the job's original due date serves as the due date until the job's slack becomes zero when its earliest finish time acts as the modified due date (Baker and Bertrand 1982). Another operation-based rule is the Modified Operation Due date (MOD) rule. The MOD priority is defined as its original ODD or its earliest finish time, whichever is larger. Baker and Kanet (1983) tested these rules under various conditions and found that MOD outperforms other rules for unweighted tardiness at high utilization. Later, Baker (1984) compared allowance based, slack-based, and ratio-based rules against the modified rules (MDD and MOD). His results showed that slack-based rules do not offer an advantage over simpler allowance-based rules, and operation-based rules perform better than job-based rules. In another study, Christy and Kanet (1990) showed that OD is the preferred rule for the mean tardiness criterion in systems with forbidden early shipment.

Another popular rule is COVERT (Cost OVER Time) which was specifically developed for the tardiness objective (Carrol 1965). The COVERT priority index represents the expected incremental tardiness cost per unit of imminent processing time. The expected tardiness is a relative measure of how much tardiness a job might experience if it is delayed by one time unit. COVERT can be converted into the weighted version since its derivation depends on the tardiness costs. In this case, COVERT includes weight as a multiplier. If job i queuing for operation j has zero or negative slack, then its expected cost per unit time is w_i and priority is w_i/p_{ij} . If its slack exceeds some worst case estimate of the remaining waiting time over remaining operations, its expected cost is set to zero. If slack is between these extremes, then the priority goes up linearly as slack decreases.

Russell *et al.* (1987) examined the sensitivity of the COVERT rule to various operating conditions and proposed different versions of COVERT. These rules were compared with EDD, SLACK, S/OPN, SPT, truncated SPT, MDD, MOD, and Apparent Urgency (AU, the very first version of ATC). The results showed that COVERT is the best rule for the mean tardiness and mean conditional tardiness criteria. Shultz (1989) proposed an expediting heuristic (CEXSPT) which attempts to lessen the undesirable properties of SPT by controlling the jobs with long processing times. Job-based and operation-based due date information is employed to expedite the late jobs. In the simulation experiments, CEXSPT and COVERT produced lower unweighted tardiness.

In a recent study, Anderson and Nyirenda (1990) developed two new rules. The first rule combines CR and SPT by assigning operation due date d_{ij} for each job i waiting for operation j as

$$d_{ij} = \max \{t + CR_{ij}(t)p_{ij}, t + p_{ij}\}.$$

The other rule combines S/RPT and SPT (S/RPT+ SPT) by assigning operation due date

$$d_{ij} = \max \{t + S/RPT_{ij}(t)p_{ij}, t + p_{ij}\} \text{ or } d_{ij} = \max \{t + (CR_{ij}(t) - 1)p_{ij}, t + p_{ij}\}.$$

It seems that these two rules are very competitive and hence they will be tested in our study.

2.2. Bottleneck dynamics studies

Early BD studies started with the development of *Apparent Tardiness Cost* (ATC) by Vepsalainen and Morton (1987). ATC is very similar to COVERT with two main differences: First, the slack is *local resource constrained slack* which takes into account the waiting times on downstream machines. Second, the decay function for the weight/processing time ratio is exponential rather than linear. In ATC, global slack is allocated to the remaining lead time, which gives local resource-constrained slack as

$$SS_{ij}(t) = d_i - \sum_{q=j+1}^{m_i} (W_{iq} + p_{iq}) - p_{ij} - t,$$

where the summation is over all unfinished downstream operations indexed as $j+1, \dots, m_i$ and W_{iq} is the estimated waiting time for operation q of job i . (The negative local slack can also be interpreted as the estimated lateness if the job is scheduled immediately at time t so that its estimated completion time is $t + p_{ij} + \sum_{q=j+1}^{m_i} (W_{iq} + p_{iq})$.)

The full priority formula is then

$$ATC_{ij}(t) = \frac{w_i}{p_{ij}} \times \exp \left(- \frac{(SS_{ij}(t))^+}{Kp_{avg}} \right),$$

where $(x)^+ = \max \{0, x\}$. In this formula, the exponential term is called the *urgency factor* if the job is scheduled and expected to be completed with slack SS_{ij} . Hence, the urgency factor is

$$U_{ij}(t) = \exp \left(- \frac{(SS_{ij}(t))^+}{Kp_{avg}} \right).$$

There are several methods to estimate waiting times. One quick and dirty method called *standard estimation* inspired from the total work content rule was proposed by the inventors of the rule: calculate W_{iq} as proportional to its processing time as $W_{iq} = bp_{iq}$, where b is a constant multiplier. One issue in this method is selecting the right multiplier value. In actual systems this can be done by using regression analysis with historically collected waiting times. In our study, we tried a couple of values for b and fixed at one level. Vepsalainen and Morton (1988) proposed two additional techniques for waiting time estimation. The method called *priority-based estimation* estimates shorter waiting times for high-priority jobs and longer for low

priority jobs. The method referred to as lead time iteration (LTI) is an iterative procedure which aims to improve waiting time estimates from one iteration to the next. The results showed an improvement in the performances of ATC and COVERT with the LTI method whereas there is no significant effect of the priority-based estimation as compared to the standard method outlined above. Another alternative might be to calculate waiting times based on queue content and length as proposed in Kanet and Zhou (1993). (We test both the standard and these queue-based methods in our study.)

Kanet and Zhou (1993) proposed a decision theory approach called MEANP. It seems that MEANP produces low unweighted tardiness and the fraction of tardy jobs as compared to FCFS, SPT, COVERT, MOD and ATC, COVERT. However, the authors recommended MOD for practical applications, since it is simple and does not require parameter estimations. In another study, Ovacik and Uzsoy (1994) extended ATC by including sequence-dependent set-up times. They compared it with several versions of EDD and ODD in a dynamic job shop environment. The results showed that the maximum lateness performance of ATC is not as good as that of ODD dispatching.

Morton *et al.* (1988) developed the early version of the Bottleneck Dynamics (BD) rule called SCHED-STAR. They used cost-benefit analysis to make scheduling decisions based on net present value (NPV) of revenues and costs. In experiments with small-size static problems, SCHED-STAR was found to be the best performer when compared with the different versions of COVERT, CR, and EXP-ET (Early/Tardy version of ATC).

In another study, Lawrence and Morton (1993b) tested resource pricing heuristics in static multi-project scheduling problems. They developed a resource price-based rule similar to SCHED-STAR and tested five pricing schemes. The results indicated that all pricing heuristics dominate the 20 benchmark rules tested. In their later work, Lawrence and Morton (1993a) tested two new resource pricing heuristics in a static job shop. Results showed that resource pricing has the potential to improve mean weighted tardiness.

Recently, Morton and Pentico (1993) brought together all the pieces of their previous studies and called it Bottleneck Dynamics (BD). BD is very similar to ATC in that it uses the numerator of the ATC function $w_i U_{ij}(t)$ as an *activity price* which is a reflection of the current scheduling decision to the weighted tardiness. In contrast to ATC, BD trades off this activity price with *total remaining resource usage* instead of current processing time. The *resource usage* of job i for operation q at machine $k(q)$ at time t is calculated as the resource price of the machine times the processing time of the operation, i.e. $R_{k(q)}(t)p_{iq}$, where $R_{k(q)}(t)$ is the resource price of the machine $k(q)$. The *total remaining resource usage* is defined over all remaining operations of the job and the ratio of activity price to this resource usage gives the BD priority:

$$BD_{ij}(t) = \frac{w_i U_{ij}(t)}{\sum_{q=j}^{w_i} R_{k(q)}(t)p_{iq}}.$$

In this way, BD prioritizes the jobs with larger activity prices (most urgent ones), while penalizing the jobs with longer processing times on bottleneck machines which presumably have high resource prices.

2.2.1. Resource pricing

From an optimization point of view, resource price can be interpreted as the value of a dual variable for the resource's capacity constraint. Hence, the resource price can be estimated by calculating extra costs if the resource capacity is decreased for one time unit. One such approach, called *empirical resource pricing*, is proposed in Lawrence and Morton (1993b). The change in objective function gives a relative measure of resource price. Since both exact calculation and the empirical pricing are difficult to implement in a dynamic environment, approximation methods have been developed.

We first consider the price of the current machine in resource usage calculation, which is referred to as *myopic pricing* in the terminology of Lawrence and Morton (1993b). In this method, the current machine's price is scaled to 1 and all others have zero prices (we called BD with myopic pricing BD-Myp; this is actually equivalent to ATC). The second alternative is *uniform pricing* (BD-Unif), which assumes that all resources are of equal importance and assigns them resource prices of 1. In this case, the denominator of the priority formula is the summation of processing times of downstream operations, P_{ij} . The third alternative, *bottleneck pricing* (BD-Bot), identifies the bottleneck machine with highest utilization, and gives it a scaled price 1.0, while other resources are assigned prices of zero.

Morton and Pentico (1993) developed other approximate methods based on busy-period analysis and queuing theory. The *static pricing* (BD-Stat) depends on the machine's utilization:

$$R_k(t) = (wU)_{\text{avg}} \frac{\rho_k}{(1.0 - \rho_k)^2},$$

where $(wU)_{\text{avg}}$ is the average activity price of the jobs and ρ_k is long term utilization rate of machine k . A more complex *dynamic price*, which changes according to the current queue length and utilization of the machine, is given by

$$R_k(t) = \sum_{i=1}^{L_k(t)} w_i U_{ij}(t) + (wU)_{\text{avg}} L_k(t) \frac{\rho_k}{1.0 - \rho_k},$$

where $L_k(t)$ is the current queue length. We will test two different versions of BD with dynamic pricing. The first one uses standard waiting time estimation (called BD-DynSt), while the other uses the estimation technique developed by Kanet and Zhou (1993) called BD-DynKZ). The waiting time for job i for operation $j(i)$ at machine k with queue length $L_k(t)$ is estimated by

$$W_{ij(i)} = \frac{\sum_{i=1, i \neq i}^{L_k(t)} p_{lj(i)}}{2}.$$

2.2.2. Inserted idleness

As pointed out previously, inserted idleness can improve the performance of the dispatching rules in a dynamic environment. However, one should know when it is worth keeping the machine idle even though it has jobs in its queue. As shown by Morton and Ramnath (1992), for any regular objective, no operation can be scheduled next on a given machine, unless it is currently available in the queue, or else will arrive before the current time plus the processing time of the shortest job in the

queue. By observing this fact, they proposed a modification of the ATC rule for the dynamic single machine problem. Assuming arrival times to the machine are known, they calculated the priorities of those jobs with a penalty proportional to inserted idleness involved. In this paper, we have extended their approach to job shops.

In our case, the arrival times are not known in advance. However, for a job that is being processed at a machine, we can exactly calculate the job's arrival time to the next machine in its routing. In this way, we extend the candidate job set by adding the jobs that will arrive in the near future. We reduce the priorities of such jobs proportional to the idleness. The proportionality multiplier β goes up linearly with the utilization of machine k as suggested in Morton and Ramnath (1992). If we denote the arrival time of job i to machine k for operation j as a_{ij} , then the priorities of such jobs are revised according to the following

$$BD_{ij}(t)' = BD_{ij}(t) \times \left(1.0 - \beta \frac{(a_{ij} - t)^+}{p_{\min}} \right),$$

where p_{\min} is the processing time of shortest job in the queue. This means that the original priority of a job is degraded by a term proportional to the idleness as a fraction of the minimum of the processing times of waiting jobs. If this priority is still greater than the priorities of the jobs in the queue, then the machine is kept idle until this job arrives at the machine. We employ this method only for BD although it is not too difficult to modify other dispatching rules to include this type of inserted idleness.

3. System considerations and experimental factors

The environment considered in this study is a dynamic shop with the classic job shop assumptions (Morton and Pentico 1993, p. 387). It is a re-entrant job shop with 10 machines continuously available. Jobs arrive according to a Poisson process and are released immediately to the system. The jobs have a number of operations uniformly distributed between 1 and 10. The operations are randomly processed through the machines. Two types of shop are simulated: the first one is a *uniform shop* where every machine is approximately equally utilized with operation processing times drawn from a uniform distribution (Anderson and Nyirenda 1990, Vepsalainen and Morton 1988). The second is a *bottleneck shop* where one or more machines are bottlenecks with longer processing times. The relative speeds of three machines are highly utilized, some are lightly utilized. Job weights are drawn from U (Anderson and Nyirenda 1990, Vepsalainen and Morton 1988).

Since our aim is not to investigate the due date setting policies, we assign more general and somewhat random due dates controlled by a processing time multiplier to generate desired levels of due date tightness (similar external setting of due dates can be found in the literature, e.g. Vepsalainen and Morton (1989). Specifically, due dates are assigned randomly over a full range of flow allowances, with a maximum of 12.0, 9 and 6.0 times the mean total job processing time for relatively loose, medium, and tight due dates, respectively. We call this rule the random average total work content rule (RATWK), formulated as follows:

$$d_i = r_i + U \left[0, D * \left(\frac{(m_{\min} + m_{\max})}{2} \right) p_{\text{avg}} \right],$$

Factor	Number of levels	Levels
Shop type	2	Uniform, bottleneck
Utilization	3	Low (60%), Medium (80%), High (90%–95%)
Due date tightness	3	Loose, Moderate, Tight
Dispatching rules	30	BD rules (6), X-BD rules (6), Others (18)

Table 2. Experimental factors and their levels in the simulation experiments.

where D is the tightness parameter, m_{\min} and m_{\max} are minimum and maximum number of operations, respectively. The average utilization of the shop is determined by calibrating the arrival rate of the jobs. In a uniform shop, the arrival rate is adjusted to achieve approximately 60% average utilization in the low level, 80% in the medium level, 90% in the high level. In the bottleneck shop, the utilization of the bottleneck determines the system load level. In this case, arrival rate is adjusted to achieve 60% utilization on the bottleneck machine in the low level, 80% in the medium level and 95% in the high level. (For a summary of experimental factors, see table 2.)

The system is simulated by using SIMAN simulation language (Pegden *et al.* 1990) with additional C subroutines linked within the UNIX environment. We use the method of batch means to collect the output data. Based on pilot runs, we determine a conservative warm-up period for the system as 2500 job completions. To reduce correlation between batches, we use 10 rather long batches with 1000 jobs each. Common random numbers are used as a variance reduction technique to provide the same experimental conditions for the rules (Law and Kelton 1991).

The dispatching rules presented in Appendix B are used in the experiments. We test 6 different pricing schemes for BD:

- (1) BD with myopic pricing (BD-Myp);
- (2) BD with dynamic pricing (standard waiting time estimation, BD-DynSt);
- (3) BD with dynamic pricing (K-Z waiting time estimation, BD-DynKZ);
- (4) BD with static pricing (BD-Stat);
- (5) BD with uniform pricing (BD-Unif);
- (6) BD with bottleneck pricing (BD-Bot).

Inserted idleness versions of these rules are represented by adding X as a prefix (X-BD-Myp, X-BD-DynSt, etc.). In the inserted idleness case, only the jobs whose next operation is on the machine under consideration are considered as additional candidates, as described in the previous section.

The previous research has indicated that the performances of parametric rules such as COVERT and BD are sensitive to the selection of the parameter value. For example, studies showed that look-ahead parameter K should be adjusted according to conditions such as utilization and due date tightness (Morton and Rachamadugu 1982, Vepsalainen and Morton 1988). We have conducted some pilot experiments to choose values for parameters. As a result of these simulation runs, COVERT is found to be robust and we use $b = 2.0$ and $h = 2.0$ for all settings. For BD rules, the waiting time estimation parameter (b) is set to 2.0. The look-ahead parameter (K) is adjusted according to the job shop type and utilization level. In the uniform shop K is fixed at 2.0 in low utilization, $K = 3.0$ in medium, $K = 4.0$ in high utilization. In the bottleneck shop, K is 1.5 in the low and medium levels of utilization, and

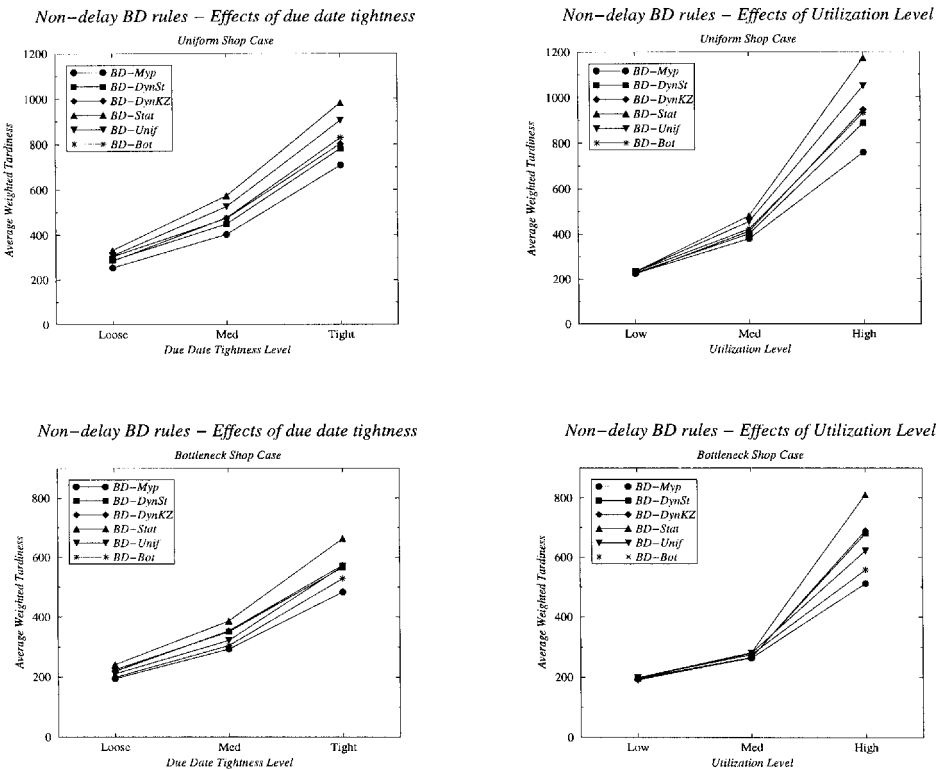


Figure 1. Effects of utilization and due date tightness on average weighted tardiness performance of non-delay BD rule with different resource pricing schemes for uniform and bottleneck shops.

it is fixed at 3.0 in high utilization. Experiments conducted by Morton and Ramnath (1992) show that $\beta = 1.3 + \rho_k$ is appropriate for ATC with inserted idleness in the single machine case. In order to find the best value for the job shop, we have tried different values for β for the inserted idleness version of BD, and finally set $\beta = 2.0 + \rho_k$.

4. Computational results

First, we measure the performance of the BD rules and identify the promising ones that can be used for further tests. Then we present the overall results of other dispatching rules.

4.1. Experiments with bottleneck dynamics

Figure 1 shows effects of due date tightness and utilization level on the weighted tardiness (T) performance of non-delay BD rules. As one can intuitively expect, the performances deteriorate as the due date tightens and/or the utilization increases. However, the sensitivity of BD to these changes varies for different resource pricing schemes. For example, while every pricing scheme yields almost the same WT at low utilization rates, their performances are different at medium and high utilization levels. In the uniform shop case, BD with myopic pricing (BD-Myp) draws the lower envelope regardless of the utilization level and due tightness. However, the

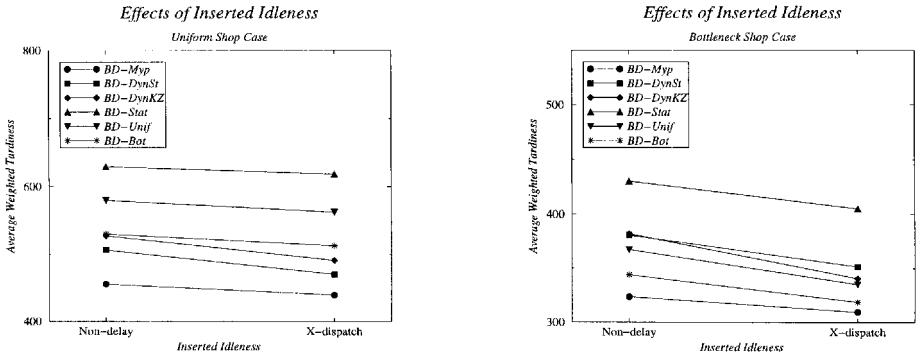


Figure 2. Effects of inserted idleness method on average weighted tardiness performance of BD rule with different resource pricing schemes.

tests conducted at the 5% significance level show that the differences among the best four BD rules (BD-Myp, BD-DynSt, BD-DynKZ and BD-Bot) are statistically insignificant in general, although they are all significantly better than BD-Stat and BD-Unif. The difference between the latter two is not distinguishable.

In the bottleneck shop, although the main observations are somewhat similar to those of the uniform case, there are some differences that show the effects of unbalanced machine loads. First, we observe that as expected the performances of the rules are less distinguishable at low and medium utilization rates than the uniform case, since the overall utilization is lower in the bottleneck case. The second best performer after BD-Myp is BD with bottleneck resource pricing (BD-Bot) instead of BD with dynamic pricing. This indicates the effectiveness of bottleneck resource pricing for unbalanced shops.

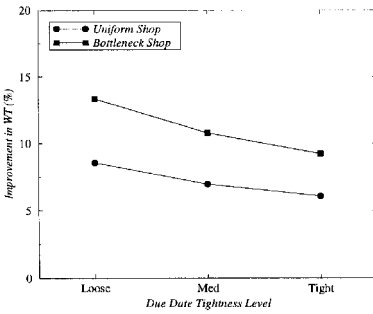
Results show that our previous observations about non-delay BD rules are still valid for X-versions of BD. For brevity, we plot WT with respect to inserted idleness considerations in order to see the overall effect of inserted idleness. As seen in figure 2, there is always improvement due to inserted idleness, although the improvement rate changes according to the pricing method and the shop type. For instance, in the uniform shop, inserted idleness considerably improves performances of both dynamic pricing methods (BD-DynSt and BD-DynKZ). The difference is almost negligible for other pricing schemes. We also note that the improvement is more significant for almost all types of pricing schemes in the bottleneck case.

In general, the improvement achieved by inserted idleness over non-delay versions varies from 1.4% to 16.8%. The overall improvement due to inserted idleness is statistically significant at the low and medium utilization levels. This improvement is especially visible for BD with dynamic pricing (BD-DynSt and BD-DynKZ). To observe the effect of inserted idleness with respect to utilization and due date tightness, we plot the percent decrease in WT of a selected BD rule (BD-DynKZ, see figure 3). It seems that the improvement in the bottleneck shop is always more than the one in the uniform case. The improvement percentage decreases with increasing utilization and tightening due dates.

We can summarize the main findings as follows.

- BD with myopic pricing performs better than the others. This interesting result can be attributed to several reasons. First, myopic pricing calculates the total

Inserted Idleness – Effects of due date tightness



Inserted Idleness – Effects of utilization

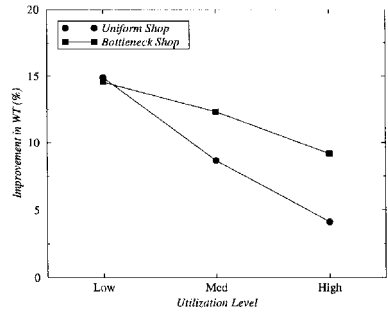


Figure 3. Effects of due date tightness and utilization on the percentage improvement in weighted tardiness achieved by implementing inserted idleness with BD-DynKZ dispatching rule.

resource usage by considering only the current machine's price scaled to 1.0 and the current operation processing time, both of which are deterministic. In global pricing schemes, the total usage cost depends on estimated prices of downstream machines. If these prices are not accurate, they might lead to inferior scheduling decisions. If we want to calculate the price of a downstream machine we should know the load of that machine when the job arrives at that machine. This is very difficult to predict in dynamic environments such as ours. As shown in the results, assuming that all machines have the same resource price (BD with uniform pricing) does not yield encouraging performance. Hence, discriminating machines using different prices improves the performance significantly. At this point, the focus should be on developing better schemes with better predicting mechanisms. Global pricing was shown to be better than myopic pricing in previous studies (Lawrence and Morton 1993b). One important difference between these studies and ours is that they used a static environment where the scheduler can estimate the prices with high accuracy.

- Inserted idleness improves the performance of BD, up to 16.8% in this research. As we have shown, the improvement is considerably higher at low utilization levels. Since there is slack in capacity and time in these cases, the scheduler has more opportunities to wait for a hot job, and the cost of keeping the machine idle is not as high as in a congested shop. Hence, inserted idleness versions of BD can be suggested for low and moderately loaded shops. However, it should be noted that in reality these are not usually the situations in which tardiness is a major problem.
- The results with the bottleneck shop are somewhat different from those of the uniform case. This is expected because the purpose of resource pricing is to discriminate the machines according to their loads. Therefore, in an unbalanced shop, resource pricing schemes effectively distinguish the resources as bottleneck and non-bottleneck. The performance of BD with bottleneck pricing is better in the bottleneck shop than in the uniform shop.

4.2. Experiments with other rules

Since the number of rules to be tested is quite large, we have put them into four groups for easy presentation of the results: (1) FCFS, WSPT, WLWKR; (2) EDD,

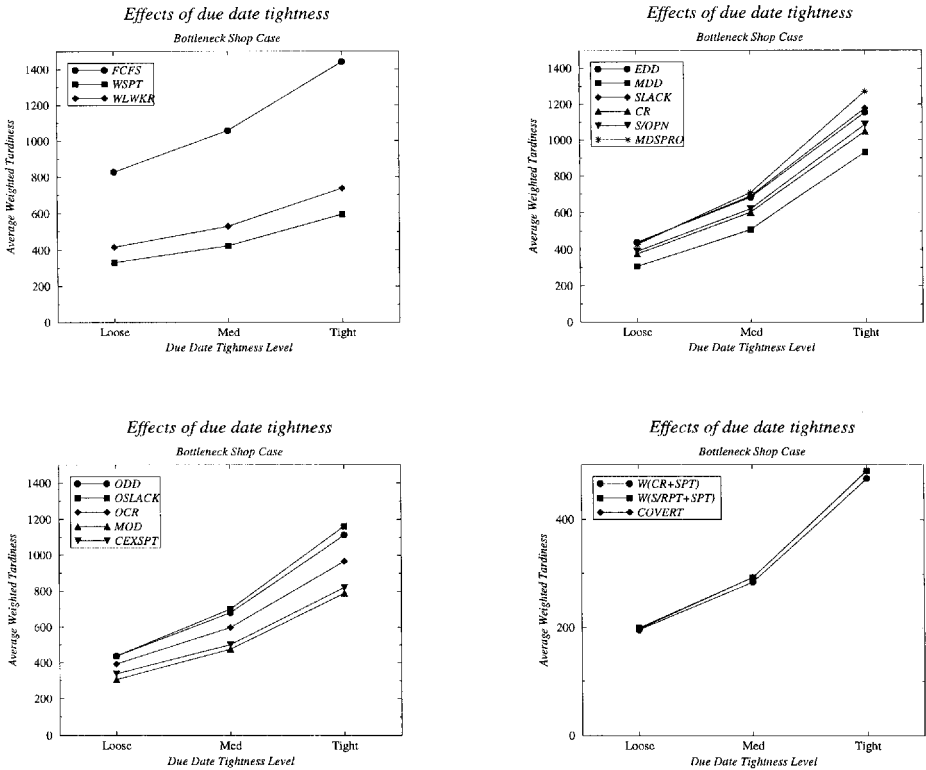


Figure 4. Effects of due date tightness on average weighted tardiness performance of the non-BD rules for bottleneck job shop.

MDD, SLACK, CD, S/OPN, MDSPRO (mostly job-based); (3) ODD, OSLACK, OCR, MOD, CEXSPT (mostly operation-based); and (4) W(CR+ SPT), W(S/RPT+ SPT), COVERT.

Relative performances of the rules are very similar in both uniform and bottleneck shops except that the differences are more visible in the bottleneck case. Therefore, we present only the bottleneck shop results for brevity. As figures 4 and 5 show, the differences between the performances of the rules become more significant with tightening due dates and increasing shop loads.

The best rule in the first group, WSPT, is most robust to changes in the levels of experimental factors whereas FCFS is the worst. MDD is the best rule in the second group, followed by CR and S/OPN. Although MDSPRO was designed to eliminate the drawbacks of S/OPN, it performs poorly. Among the operation-based rules in the third group, MOD and CEXSPT compete for the first rank, while the operational slack rule (OSLACK) draws the upper envelope. The last group includes the best performers among all the rules tested. The maximum difference between the worst and the best performer in this group is around 4.3% in the uniform shop and 2.8% in the bottleneck case. It is difficult to rank them since their performances are close to each other. The statistical tests conducted on the overall results show that the rules in the last group (W(CR+ SPT), W(S/RPT+ SPT) and COVERT) are all significantly better at the 5% significance level than the others. However, the performance differences within the last group are indistinguishable. If we group the

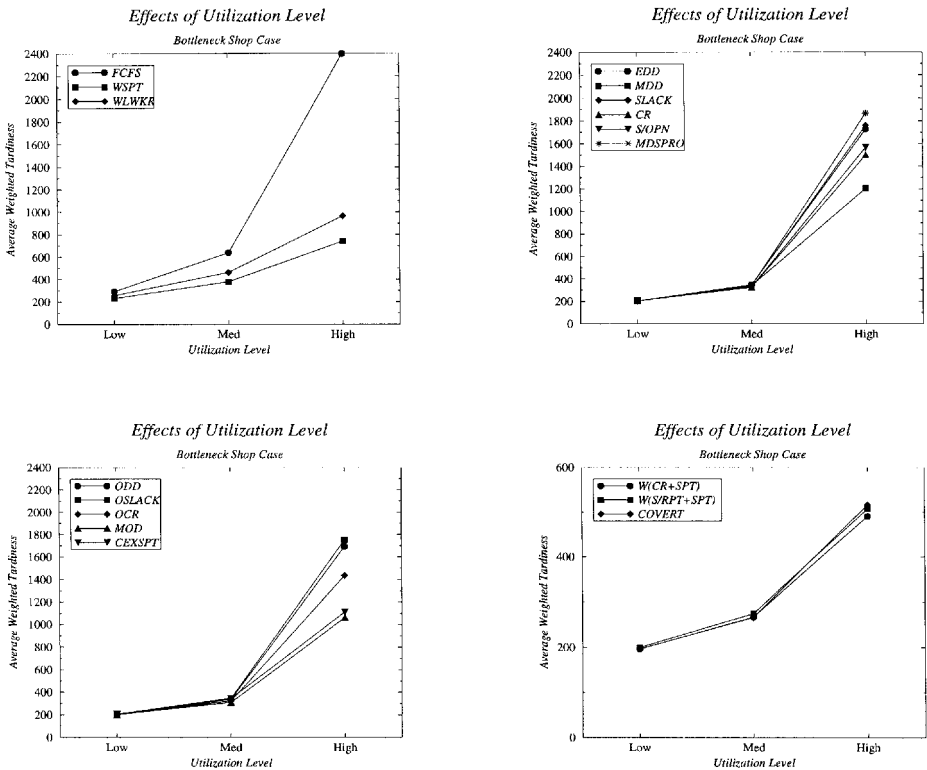


Figure 5. Effects of utilization on average weighted tardiness performances of the non-BD rules for bottleneck job shop.

rules according to their statistically significant difference, we achieve the following ranking: (1) W(CR+ SPT), W(S/RPT+ SPT) and COVERT, (2) WSPT, (3) MOD, (R) CEXSPT, WLWKR, OCR, (5) CR, MDD, ODD, S/OPN, OSLACK, (6) SLACK, EDD, MDSPRO, (7) FCFS. The low ranked groups are better than the higher ranked groups and the overall performance difference within a group is not significant.

The major conclusions in this section can be summarized as follows.

- Priority rules which make use of operational information such as operation due dates and operation processing times perform consistently better than their job-based counterparts. The results also show that the performances of the job-based and operation-based rules are very sensitive to the system conditions.
- WSPT is more robust in general. For instance, while WSPT is worse than almost all job- and operation-based rules at low load levels, it performs better than these rules in congested shops with tight due dates.
- The rules in the last group perform quite well regardless of changes in due date tightness and utilization levels. Complex and composite rules such as COVERT and W(CR+ SPT) are very effective when compared with the other rules. Although MOD is reported as the best rule for unweighted tardi-

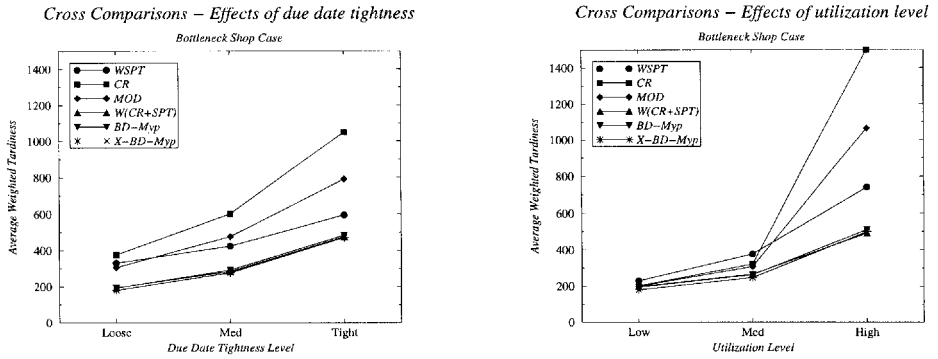


Figure 6. Effects of due date tightness and utilization level on weighted tardiness performances of selected rules.

ness in previous studies, it is outperformed by these new rules tested in our study.

4.3. Cross comparisons of BD and other rules

Finally, we compare the best performing BD rules (non-delay and X-versions of BD with myopic pricing) and selected rules WSPT, CR, MOD and W(CR+ SPT). For the same of brevity, we present the results for the bottleneck shop. As depicted in figure 6, the BD rules are quite effective in improving the WT performance when compared to all the rules except W(CR+ SPT). The tests show that the overall performances of the best four BD heuristics and the best three among conventional rules are not statistically significant. This shows that W(CR+ SPT) is very successful in combining the characteristics of CR and SPT into one rule. Note that CR is relatively good at low load levels/loose due dates, whereas WSPT produces very good WT results at high utilization rates/tight due dates. This rule discriminates jobs with different due dates and processing time requirements. This is similar to the urgency factor idea behind the BD rules, but BD also discriminates between the resources according to their loads. In our experiments, however, we did not observe any additional benefits of pricing over the best non-parametric rule (W(CR+ SPT)).

To get more insight, we have collected percent tardy (PT) and conditional weighted tardiness (CWT) statistics. As shown in figure 7, BD is a good performer in PT regardless of due date tightness and utilization level. Since its WT performance is close to that of W(CR+ SPT), it is expected that BD yields poor CWT performances as compared to W(CR+ SPT) (see figure 8). In general, BD loses its advantage by producing jobs with long tardiness even though tardy jobs are considerably less. MOD has a similar drawback. We note that W(CR+ SPT) is the best in CWT in almost every condition. Our previous observation about W(CR+ SPT) in combining WSPT and CR is also strengthened with these graphs. The CWT measure of WSPT is very high with loose due dates and low utilization levels, whereas CR is relatively good in that region. As utilization increases or due dates tighten, WSPT's curve intersects CR's curve and WSPT starts producing less CWT than CR does. Hence, we see that W(CR+ SPT) combines the advantageous parts of the two rules.

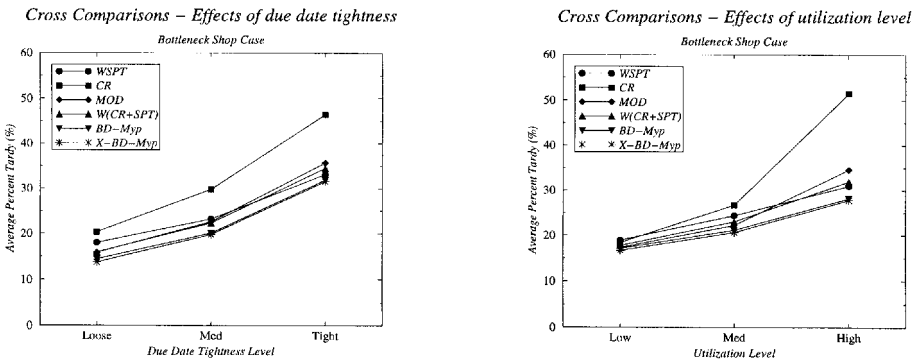


Figure 7. Effects of due date tightness and utilization on percent tardy performance of selected rules in bottleneck shop case.

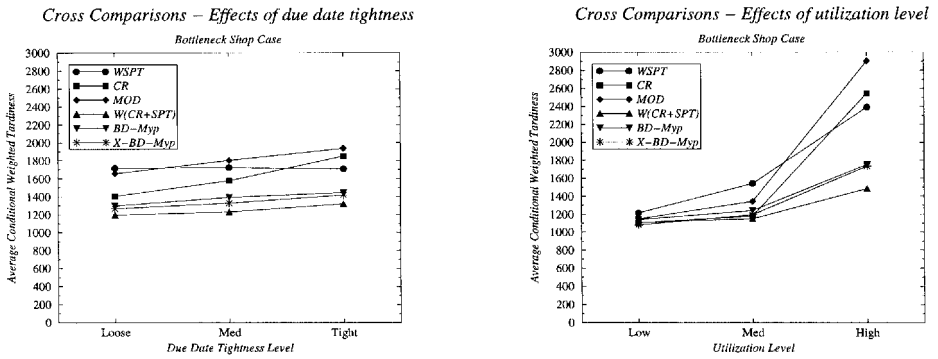


Figure 8. Effects of due date tightness and utilization on conditional weighted tardiness performance of selected rules in bottleneck shop case.

5. Conclusions and further research directions

In this study, we measured the performance of numerous dispatch heuristics using simulation. We also tested new scheduling techniques, such as dispatching with inserted idleness and resource pricing. The results indicated that the priority rules which make use of operational information such as operation due dates and operation processing times perform consistently better than their job-based counterparts. Additionally, we observed that complex and composite rules designed specifically for tardiness, such as W(CR+ SPT), COVERT and BD, are very effective in reducing weighted tardiness. We also tested several resource pricing heuristics for BD and found that different pricing schemes should be used in different environments. In general, myopic pricing is very effective. In unbalanced shops, however, bottleneck resource pricing can also be suggested.

The results show that inserted idleness improves the performance of ordinary dispatching heuristics. The inserted idleness versions of BD can be especially suggested for low and medium utilization rates. Actually, inserted idleness would have improved the performance more if we had expanded the candidate job set. In this study, we considered only the jobs that are being processed and those whose next

operations are on the machine under consideration. To achieve further improvement, all the jobs in the system must be considered and the arrival times of these jobs to the machine should be accurately estimated. We consider this a good future research topic.

The results also indicated that there are very promising tardiness related rules such as W(CR+ SPT) and COVERT. Although BD outperforms many other dispatching rules, we could not show that it can beat W(CR+ SPT) and COVERT. One reason for this could be the resource pricing scheme utilized in BD. More effort should be expanded to develop effective resource pricing schemes in a dynamic environment. A second reason may be the urgency factor calculation, which is based on estimated waiting times and a look-ahead parameter. For the look-ahead parameter, we can decide on the value by making some pilot runs. However, as Morton and Pentico (1993) noted, the lead time estimation is one of the important issues for better performance. If the lead time is accurately estimated, then the scheduler can predict estimated lateness of a job very well. Hence researchers need to find more effective ways of estimating waiting times or total lead time.

In this study, we tested Kanet and Zhou's (1993) waiting time estimation whose performance is good but not sufficient to get higher-quality performances. Another option is the multi-pass version of BD known as lead time iteration (LTI). We did not include LTI in our study because of the reasons outlined previously. Further research can focus on LTI and other methods of estimating lead times in a dynamic job shop. Additional improvement for BD might be achieved by analysing why the rule yields few but long tardy jobs (high conditional weighted tardiness).

Another further research topic is testing the rules using different order review release (ORR) methods. In our study, we used immediate release, but it would be very interesting to see the performance of sophisticated rules under other ORR policies.

We also note that up-to-date accurate information need is highest for BD. The majority of the rules in our test set are dynamic, which require recalculation of the priority as time advances. However, they utilize mostly job-related attributes which might be readily available in most of the real systems. In contrast, the rules such as COVERT and BD need more shop status information (machine utilization, queue length, etc.) which can be gathered only in sophisticated shops with a good information collection/distribution system. If the shop lacks such an information network, then it is better to use less-demanding rules such as W(CR+ SPT). On the other hand, if such a network is already available, which is very viable with today's hardware and software infrastructure, then BD or other information-based rules can easily be employed. In this case, historical data can be used for value estimation in BD and other parametric rules (utilization, waiting time, etc.).

Acknowledgment

We thank the anonymous referees for providing us with constructive comments which improved the initial draft.

Appendix A: Notation and symbols used in the paper

- i job index
- j, q operation index
- $j(i)$ operation j of job i

- k machine index
 t current time
 n number of jobs in a specific planning horizon or in a simulation run
 m_i number of operations of job i
 w_i weight or tardiness penalty of job i
 d_i due date of job i
 C_i completion time of job i
 T_i tardiness of job i , $\max \{0, C_i - d_i\}$
 a_i initial flow allowance of job i
 $A_i(t)$ flow allowance of job i at time t
 a_{ij} arrival time of job i for operation j to the current machine
 p_{ij} processing time of operation j of job i
 p_{avg} average processing time of jobs waiting for a resource
 p_{ij} remaining total processing time of job i from operation j
 $S_{ij}(t)$ slack of job i waiting for operation j at time t
 m_{ij} remaining number of operations in job i from operation j
 d_{ij} operation due date of operation j of job i
 W_{ij} estimated waiting time for operation j of job i
 TL_{ij} estimated tail lead time for operation j of job i
 $SS_{ij}(t)$ local resource constrained slack of operation j of job i at time t
 $U_{ij}(t)$ urgency factor of operation j of job i at time t
 $AP_{ij}(t)$ activity price of operation j of job i at time t
 b waiting time estimation multiplier
 K, h look-ahead parameters
 $k(q)$ the machine required for operation q of the job under consideration
 $R_k(t)$ resource price of machine k at time t
 ρ_k utilization of machine k
 $L_k(t)$ queue length of machine k at time t
 $p_{\text{min } k}$ minimum processing time of operations waiting for machine k
 β inserted idleness parameter
 WT average weighted tardiness measure $WT = (\sum_{i=1}^n w_i T_i) / n$
 PT per cent tardy measure, $PT = [(\sum_{i=1}^n \delta(T_i) / n) \times 100]$
 $\delta(T_i) = \begin{cases} 1.0 & \text{if } T_i > 0 \\ 0.0 & \text{otherwise} \end{cases}$
 CWT conditional weighted tardiness measure:

$$CWT = \frac{\sum_{i=1}^n w_i T_i}{(PT \times n / 100)}.$$

$$(x)^+ \quad \max \{0, x\}$$

Appendix B: Priority dispatching rules

The priorities are calculated for job i waiting for operation j at machine k at time t as listed in the following table.

Priority rule	Description
FCFS (first come first served)	$FCFS_{ij} = a_{ij}$
WSPT (weighted shortest processing time)	$WSPT_{ij} = \frac{w_i}{p_{ij}}$
WLWKR (weighted least work remaining)	$WLWKR_{ij} = \frac{w_i}{\sum_{q=j}^{m_i} p_{iq}}$
EDD (earliest due date)	$EDD_i = d_i$
MDD (modified due date)	$MDD_{ij}(t) = \max \left\{ d_i, t + \sum_{q=j}^{m_i} p_{iq} \right\}$
SLACK (least slack)	$SLACK_{ij}(t) = d_i - t - \sum_{q=j}^{m_i} p_{iq}$
CR (critical ratio)	$CR_{ij}(t) = \frac{d_i - t}{\sum_{q=j}^{m_i} p_{iq}}$
S/RPT (slack per remaining processing time)	$S/RPT_{ij}(t) = \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{\sum_{q=j}^{m_i} p_{iq}}$
S/OPN (slack per remaining operation)	$S/OPN_{ij}(t) = \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{m_i - j + 1}$
MDSPRO (modified dynamic slack per remaining operation)	$MDSPRO_{ij}(t) = \begin{cases} \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{m_i - j + 1} & \text{if } d_i - t - \sum_{q=j}^{m_i} p_{iq} > 0, \\ (m_i - j + 1) \left(d_i - t - \sum_{q=j}^{m_i} p_{iq} \right) & \text{otherwise} \end{cases}$
ODD (operation due date)	$ODD_{ij} = r_i + \frac{d_i - r_i}{\sum_{q=j}^{m_i} p_{iq}} \times \sum_{q=1}^j p_{iq}$

OSLACK
(operation slack)

$$OSLACK_{ij}(t) = r_i + \frac{d_i - r_i}{\sum_{q=j}^{m_i} p_{iq}} \times \sum_{q=1}^j p_{iq} - t - p_{ij}$$

OCR
(operation critical ratio)

$$OCR_{ij}(t) = \frac{r_i + \frac{d_i - r_i}{\sum_{q=j}^{m_i} p_{iq}} \times \sum_{q=1}^j p_{iq} - t}{p_{ij}}$$

MOD
(modified operation due date)

$$MOD_{ij}(t) = \max \left\{ r_i + \frac{d_i - r_i}{\sum_{q=j}^{m_i} p_{iq}} \times \sum_{q=1}^j p_{iq}, t + p_{iq} \right\}$$

CEXSPT (see Note 1)

W(CR+ SPT)
(weighted CR and SPT)

$$W(CR + SPT)_{ij}(t) = \frac{w_i}{p_{ij} \times \max \left\{ 1.0, \frac{d_i - t}{\sum_{q=j}^{m_i} p_{iq}} \right\}}$$

W(S/RPT+ SPT)
(weighted S/RPT and SPT)

$$W(S/RPT + SPT)_{ij}(t) = \frac{w_i}{p_{ij} \times \max \left\{ 1.0, \frac{d_i - t - \sum_{q=j}^{m_i} p_{iq}}{\sum_{q=j}^{m_i} p_{iq}} \right\}}$$

COVERT
(cost over time)

$$COVERT_{ij}(t) = \frac{w_i}{p_{ij}} \times \left(\frac{h \sum_{q=j}^{m_i} W_{iq} - \left(d_i - t - \sum_{q=j}^{m_i} p_{iq} \right)}{h \sum_{q=j}^{m_i} W_{iq}} \right)$$

BD
(bottleneck dynamics)

$$BD_{ij}(t) = \frac{w_i \exp \left(- \frac{\left(d_i - \sum_{q=j+1}^{m_i} (W_{iq} + p_{iq}) - p_{iq} - t \right)^+}{Kp_{avg}} \right)}{\sum_{q=j}^{m_i} R_k(q)p - iq}$$

Note. CEXSPT rule partitions the original queue into 3 queues which are late queue, i.e. $S_{ij}(t) = d_i - t - \sum_{q=j}^{m_i} p_{iq} < 0$, operationally late queue (behind the schedule), i.e. $O_{ij}(t) = d_{ij} - t - p_{ij} = r_i + (d_i r_i) / \sum_{q=j}^{m_i} p_{iq} \times \sum_{q=1}^j p_{iq} - t - p_{ij} < 0$, and ahead of schedule queue, i.e. $O_{ij}(t) \geq 0$. Then the rule selects SPT job from queue 1, if this job does not create a new late job with $S_{ij}(t) < 0$. If it does, then a new SPT job is selected from queue 2, if it does not create a new operationally late job in queue 3. If it does, then a new SPT job is selected from queue 3.

References

- ANDERSON, E. J. and NYIRENDA, J. C., 1990, Two new rules to minimize tardiness in a job shop. *International Journal of Production Research*, **28** (12), 2277–2292.
- BAKER, K. R., 1974, *Introduction to Sequencing and Scheduling* (New York: Wiley).
- BAKER, K. R., 1984, Sequencing rules and due date assignments in a job shop. *Management Science*, **30**, 1093–1104.
- BAKER, K. R. and BERTRAND, J. W. M., 1982, A dynamic priority rule for sequencing against due dates. *Journal of Operations Management*, **3**, 37–42.
- BAKER, K. R. and KANET, J. J., 1983, Job shop scheduling with modified due dates. *Journal of Operations Management*, **4**, 11–22.
- CARROL, D. C., 1965, Heuristic sequencing of jobs with single and multiple components. PhD thesis, Sloan School of Management, Massachusetts Institute of Technology.
- CHRISTY, D. P. and KANET, J. J., 1990, Manufacturing systems with forbidden early shipment: implications for choice of scheduling rules. *International Journal of Production Research*, **28**, 91–100.
- ELVERS, D. A., 1973, Job shop dispatching using various due date setting criteria. *Production and Inventory Management*, **14**, 62–69.
- ELVERS, D. A. and TAUBE, L. R., 1983, Time completion for various dispatching rules in job shops. *OMEGA*, **11**, 81–89.
- GERE, JR, W. S., 1966, Heuristics in job shop scheduling. *Management Science*, **13**, 167–190.
- KANET, J. J., 1982, On anomalies in dynamic ratio type scheduling rules: a clarifying analysis. *Management Science*, **28**, 1337–1341.
- KANET, J. J. and HAYYA, J. C., 1982, Priority dispatching with operation due dates in a job shop. *Journal of Operations Management*, **2**, 155–163.
- KANET, J. J. and ZHOU, Z., 1993, A decision theory approach to priority dispatching for job shop scheduling. *Production and Operations Management*, **2**, 2–14.
- LAW, A. M. and KELTON, W. D., 1991, *Simulation Modeling and Analysis* (New York: McGraw-Hill).
- LAWRENCE, S. R. and MORTON, T. E., 1993a, Myopic dispatch scheduling and bottleneck dynamics. Technical Report 1993–22, Graduate School of Industrial Administration, Carnegie-Mellon University.
- LAWRENCE, S. R. and MORTON, T. E., 1993b, Resource-constrained multi-project scheduling with tardy costs: comparing myopic, bottleneck, and resource pricing heuristics. *European Journal of Operational Research*, **64**, 168–187.
- LENSTRA, J. K. and RINNOOY KAN, A. H. G., 1978, Computational complexity of scheduling under precedence constraints. *Operations Research*, **26** (1), 22–35.
- MIYAZAKI, S., 1981, Combined scheduling system for reducing job tardiness in a job shop. *International Journal of Production Research*, **19**, 201–211.
- MORTON, T. E., LAWRENCE, S. R., RAJAGOPOLON, S. and KEKRE, S., 1988, SCHED-STAR: a price based shop scheduling module. *Journal of Manufacturing and Operations Management*, **1**, 131–181.
- MORTON, T. E. and PENTICO, D., 1993, *Heuristic Scheduling Systems with Applications to Production and Project Management* (New York: Wiley).
- MORTON, T. E. and RACHAMADUGU, R. M. V., 1982, Myopic heuristics for the single machine weighted tardiness problem. Technical Report 30-82-83, Graduate School of Industrial Administration, Carnegie-Mellon University.
- MORTON, T. E. and RAMNATH, P., 1992, Guided forward tabu/beam search for scheduling very large dynamic job shops, I. Technical Report 1992-47, Graduate School of Industrial Administration, Carnegie-Mellon University.
- OVACIK, I. M. and UZSOY, R., 1994, Exploiting real-time shop floor status information to schedule complex job shops. *Journal of Manufacturing Systems*, **13**, 73–84.
- PANWALKER, S. S. and ISKANDER, W., 1977, A survey of scheduling rules. *Operations Research*, **25**, 45–61.
- PEGDEN, C. D., SHANNON, R. E. and SADOWSKI, R. P., 1990, *Introduction to Simulation Using SIMAN* (New York: McGraw-Hill).
- RAMASESH, R., 1990, Dynamic job shop scheduling: a survey of simulation research. *OMEGA*, **18**, 43–57.

- RUSSELL, R. S., DAR-EL, E. M. and TAYLOR II, B. W., 1987, A comparative analysis of the COVERT job sequencing rule using various shop performance measures. *International Journal of Production Research*, **25**, 1523–1539.
- SHULTZ, C. R., 1989, An expediting heuristic for the shortest processing time dispatching rule. *International Journal of Production Research*, **27**, 31–41.
- VEPSALAINEN, A. P. J. and MORTON, T. E., 1987, Priority rules for job shops with weighted tardiness costs. *Management Science*, **33**, 1035–1047.
- 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.
- WEEKS, J. K., 1979, A simulation study of predictable due dates. *Management Science*, **25**, 363–373.