The effect of inserted idle time on the performance of dispatching rules in a flowshop.

Ahmed El-Bouri
Department of Operations Management & Business Statistics
Sultan Qaboos University
PO Box 20, PC 123, Muscat, Oman

Abstract

Inserted idle time occurs when a machine is kept waiting deliberately for the arrival of a job that is still being processed on a preceding machine. A common assumption in most of the existing literature on flowshop scheduling is that inserted idle time is not allowed, and whenever there are jobs that are ready for processing on a given machine, that machine is not kept idle for any reason. This study examines the effects of inserted idle times on the performance of a selection of dispatching rules that are commonly used in flowshops, and for the separate objectives of minimizing the mean flowtime, mean tardiness, or number of tardy jobs. In general, idle times are not looked upon favorably in production scheduling because they lengthen production runs and decrease machine utilizations. However, this study finds that there are benefits to insertion of short idle times under some conditions, and that the duration of these idle times depends on the shop congestion levels, the tightness of the job due dates, and the performance objective under consideration. In particular, inserted idle time is appears to be most effective in minimizing the number of tardy jobs in situations where due-dates are tight.

Keywords
Dynamic flowshop, Dispatching rule, Performance measure, Inserted idle time

1. Introduction

A flowshop is a production system that consists of *m* number of machines or processors arranged in series. Jobs that are processed in a flowshop follow a common route through the shop, beginning with the first machine and ending at the m^{th} machine, visiting each of the machines only once and in the same order. Although all jobs follow the same route, the processing time requirements for their operations on each of the machines may differ. Flowshops have been the subject of much research in production scheduling, where the problem is usually to decide in what order to process the jobs on each of the machines, so that some stated performance objective is best satisfied.

The type of flowshop considered in this paper is a dynamic one, in the sense that jobs continuously arrive while other, previously arrived jobs, are being processed. This is in contrast to a static flowshop, where all of the jobs arrive simultaneously at the start of the production run. After completion of a job's operations on one machine, the job is then transferred to the next machine in sequence. If that machine is occupied with another job, the transferred job is deposited in queue at a buffer to wait for processing. It is assumed that buffer space is always available to hold any number of waiting jobs at any machine in the flowshop.

The scheduling objective is to sequence the processing of the waiting jobs on the machines so that some overall performance measure is optimized. There are a number of measures that are often considered, and this paper investigates, as single objectives, the minimization of three such measures. These are mean flowtime, mean tardiness, and the number of tardy jobs. These measures usually conflict with one another, in that minimizing one of them occurs at the expense of one or both of the others.

The scheduling methodology investigated in this study uses dispatching rules. A dispatching rule is used for prioritizing waiting jobs at a machine. Whenever a machine is ready to be loaded with a new job, the dispatching rule decides which of the waiting jobs is selected. Dispatching rules are favoured in dynamic environments, because they can be applied in real-time, without the need to re-schedule the entire flowshop with every new job arrival. Most of the research and applications of dispatching rules assume that a machine will not be kept idle as long as there are any jobs ready and available to be processed on it. That is, inserted idle time is not allowed. However, there may be instances where the performance objectives may be better satisfied by keeping a machine deliberately idle, even though there are other jobs ready to be processed, in order to wait for a more ‘critical’ job that has yet to arrive.

The main purpose of this paper is to investigate the effect of inserted idle time on the performance of some of the best known dispatching rules in satisfying different performance objectives for the *m*-machine flowshop with dynamic arrivals. Following the notation described in Pinedo (2002), the types of flowshop problems considered in this paper are identified as $F_m/r_j/\sum C_j$, $F_m/r_j/\sum T_j$, $F_m/r_j/\sum U_j$. It is assumed that the release times symbolized by r_j are random and not known in advance at the start of the production run.

2. Literature Review

The use of inserted idle time in production scheduling has received rather limited attention over the past 50 years, since Giffler and Thompson (1960) considered inserted idle times in their study of active schedules for static jobshops. Conway et al. (1967) proved that inserted idle time is not necessary in the static single machine sequencing problem with regular performance measures (a non-decreasing function of job completion times). Since then, most of the research involving inserted idle time has centered around single machine problems with either dynamic job arrivals, non-regular performance measures, or both. A thorough survey of these may be found in Kanet and Sridharan (2000). More recent studies have continued to concentrate on the single machine problem, particularly with earliness and tardiness penalties such as in Valente and Schaller (2009) and Soroush (2010).

Kanet and Sridharan (2000) classified production scheduling problems into eight categories based on job availability at the flowshop (simultaneous or not), number of machines (single or multiple) and performance measure (regular or not). In the first category (single machine, simultaneous arrival of all jobs and regular performance measure) non-delay schedules are optimal and, as mentioned, inserted idle time need not be considered (Conway 1967). However, inserted idle times can produce better schedules in the remaining seven categories. Therefore, multi-machine shops such as the flowshop considered in this paper, may benefit from inserted idle times regardless of the performance measure being regular or not, and whether or not the job arrivals are simultaneous.

The investigation of inserted idle times in *multi-machine* production scheduling is rather sparse in the literature, despite the potential for improved scheduling. What few papers have been published deal primarily with jobshops. Carrol (1965) tested some heuristics using inserted idle time in conjunction with his COVERT dispatching rule for the weighted tardiness objective in jobshops. Morton and Ramnath (1992) proposed a TABU and beam search method for large dynamic jobshops with allowance of inserted idle times. Storer and Avci (2004) presented several heuristic solution methods, including Lagrangian relaxation and local search algorithms based on a new neighborhood structure for the total weighted earliness and tardiness job shop problem. More recently, He and Sun (2013) put forth a new idle time insertion strategy for flexible job shops that are subject to machine breakdowns.

Studies involving inserted idle time in flowshop environments are quite rare. Yoon and Ventura (2002) presented linear programming formulations to find optimal starting and completion times for all the jobs in a given sequence, including inserted idle times, for minimizing the mean weighted absolute deviation from due dates in a lot-streaming flowshop. Luo et al. (2013) investigated a hybrid flowshop with family setup times, and concluded that inserting idle times into a non-delay schedule can further reduce the total setup time as well as makespan. They also proposed a mechanism to determine the location and duration of the idle times in the schedules.

In summary, there are no known publications that specifically examine the behavior of dispatching rules with inserted idle times in a traditional *m*-machine flowshop. This paper aims to fill this gap by investigating what effect, if any, inserted idle times have on various dispatching rules that are commonly applied in flowshop scheduling. A methodology for allowing dispatching rules to insert idle times, along with a computational analysis, are outlined in the next section. This is followed in the final section by a discussion of the results, together with conclusions and recommendations related to further research on the subject .

3. Methodology

The flowshop under consideration comprises *m* machines, and jobs are assumed to arrive continuously at random points in time over the scheduling horizon. The processing time required by job *j* on machine *k* is p_{jk}. The scheduling problem is to decide in what order to process the jobs on each of the machines, such that the desired performance measure (mean flowtime, mean tardiness or number of tardy jobs is minimized).

Let
r_j = arrival time of job j.
D_j = due-date for job j.
C_j = completion time of job j's final operation.
N = total number of completed jobs.

The mean flowtime is given by:

$$\bar{F} = \frac{1}{N}\sum_{j=1}^{N}(C_j - r_j) \tag{1}$$

The mean tardiness is given by:

$$\bar{T} = \frac{1}{N}\sum_{j=1}^{N}\left(C_j - D_j\right)^{*} \tag{2}$$

Where $(C_j - D_j)^{*} = C_j - D_j$ if $C_j > D_j$; 0 otherwise.

The total number of tardy jobs is:

$$U = \sum_{j=1}^{N}(C_j - D_j)^{+} \tag{3}$$

Where $(C_j - D_j)^{+} = 1$ if $C_j > D_j$; 0 otherwise.

Previous studies covering dispatching rules in flowshops and jobshops have tended to show that certain dispatching rules are preferable to others depending on the performance objective and shop conditions, such as congestion and due-date tightness. A number of the best performing rules identified in previous literature are selected for this study, with the purpose of examining the extent to which their performances are affected by the enforcement of inserted idle times.

The dispatching rules considered in this study are the COVERT rule (Carrol 1966), crtitical ratio (CR) rule (Morton and Pentico 1993), earliest due-date first (EDD), least total work remaining (LWKR), modified due-date (MDD) and modified operation due-date (MOD) rules (Baker and Kanet 1983), shortest processing time first (SPT) rule (Baker 1974) and the total weighted processing time (TWPT) rule proposed in Rajendran (1993). The choice of these dispatching rules is based on their reported performances in the previous literature for the various performance objectives (see Blackstone et al. 1982, Haupt 1989, Rajendran and Holthaus, 1999).

The version of the COVERT rule adopted here is based on the unweighted COVERT rule described in Morton and Pentico (1993), with a value of 1.0 for the constant h in equation (4):

$$\pi_{jk} = \frac{1}{p_{ik}}\left[1.0 - \frac{(d_j - t - l_{jk})^{+}}{h l_{jk}}\right]^{+} \tag{4}$$

where π_{jk} is the priority of job j at machine k, l_{jk} is the sum of the processing times for job j from machine k to machine m, and t is the current time.

There is not one dispatching rule that dominates others for any of the performance criteria. Typically, one rule which is very effective under some conditions may perform worse than other rules under a different set of shop floor conditions. For the mean flowtime objective, the SPT, LWKR and TWPT rules are frequently cited as the better alternatives (Haupt 1989, Rajendran and Holthaus 1999).

For the tardiness-based criteria of mean tardiness and number of tardy jobs, the dispatching rules that are selected in this study are based on their reported performances in previously published studies. The choice of the MDD rule in is due to the findings reported by Kim (1992) in minimizing the total job tardiness. The MOD rule was shown by Baker and Kanet (1983) to be superior to the MDD rule for mean tardiness when due-dates are relatively tight. Rajendran and Holthaus (1999) reported good results with the COVERT rule in a 10-machine flowshop. In addition to these due-date based dispatching rules, the shortest processing time (SPT) rule has also been reported very effective for mean tardiness and number of tardy jobs when the due-dates are very tight or the shop is highly congested (Rajendran and holthaus 1999, El-Bouri 2012).

3.1 Inserted idle time

Traditionally, a dispatching rule is used to select one of the *waiting* jobs to process next. In this study, a modification is made so that the job in progress, if any, at the immediately preceding machine is also considered as a dispatching candidate. If a dispatching decision is called for at machine k (k >1), then any job j in progress on machine (k-1) will also be considered. The dispatching rule assumes that this job j is available for dispatching at machine k, but with a processing time requirement equal to p_{jk} plus the remaining time to its completion on machine k-1. If the dispatching rule does end up selecting this job j, then machine k will stay idle and wait for job j to complete its operations on machine k-1, before starting to process it directly upon its arrival. In this fashion, it is the dispatching rule which decides whether or not idle time is to be inserted.

3.2 Computational Analysis

The effects of allowing the dispatching rules to insert idle times in the manner described above are investigated by simulating operations on a five-machine flowshop, with dynamic job arrivals, and recording the performance of the different dispatching rules in regard of the three measures: mean flowtime, mean tardiness and number of tardy jobs.

A set of twenty randomly generated test problems are used in the computational analysis. Each of these test problems consists of 2000 jobs that arrive non-simultaneously to the flowshop at random time intervals. The processing time requirements, p_{jk}, for each job j on each machine k are generated from the uniform distribution U[1,99]. The dispatching rules are tested under three levels of shop congestion (low, medium and high), and four levels of due-date tightness. Hence the test problems are run in twelve separate sets, each set characterized by a the combination of congestion and due-date tightness levels employed in generating the set's test problems.

The arrival times for the jobs in a test problem are drawn from a Poisson distributed arrival process with mean arrival rate λ. The values of λ are λ=0.017, λ=0.018, and λ=0.019 arrivals per unit time, corresponding to low, medium and high shop congestion. Each job is also assigned a due-date by using the Total Work Content method (Baker and Bertrand 1981), with the tightness of these due-dates controlled by a multiple, Z, of the sum of the job's processing times on the five machines. Four values of Z (2, 4, 6 and 8) are used, representing due-date tightness ranging from tight to loose respectively. Thus, each of the twelve test sets represents a specific combination of shop congestion and due-date tightness, ranging from a high-congestion, tight due-date flowshop (λ=0.019 and Z=2) to a low-congestion, loose due-date flowshop ((λ=0.017 and Z=8).

In the computational analysis, every test set is run in multiple replications, each replication being distinguished by the limit on the amount of inserted idle time that is permitted. The need to run the tests with various limits on inserted idle time arises because it is observed the performance of the dispatching rules tends to deteriorate sharply, in most cases, in the absence of a cap on the amount of inserted idle time. The majority of the test sets are therefore run in twelve replications, with limits on the amount of inserted idle time ranging from zero in the first replication to 10 time units in the eleventh replication. The twelfth replication is run without any limit on inserted idle time. In some of the test sets, where it is observed that better performance can still be achieved with further relaxation of the limit on the inserted idle time, the number of replications is extended to twenty-six, including the unlimited idle time scenario. The mean flowtime, mean tardiness and number of tardy jobs are recorded for each replication.

The computational results from the flowshop simulations are given in Table 1 for the mean flowtime objective by using the dispatching rules SPT, LWKR and TWPT. Mean flowtimes obtained by the other dispatching rules are not presented in this Table because those rules are chiefly due date-based and not designed with any purpose of flowtime minimization. Table 2 provides the results for all eight dispatching rules with respect to the mean tardiness performance measure, while similar results for the number of tardy jobs may be observed in Table 3.

Table 1 provides the average mean flowtimes observed in the 20-problem test set under the three different arrival rates that correspond to high, medium and low shop congestion. For each of these runs, the Table provides four figures. The first, under the column headed ‘No IIT limit’ is the average mean flowtime recorded when there is no limit on the amount of inserted idle time. The second figure under the column headed ‘IIT=0’ provides the average mean flowtime when inserted idle time is not permitted at all. The third figure under the column headed ‘Limited IIT’ displays the lowest average mean flowtime obtained from the multiple replications with different caps on the amount of idle time insertion. The value of the cap that produces this best performance is indicated under the column headed ‘Cap’. The mean flowtimes obtained by using the other dispatching rules are not included in Table 1, because those rules are designed primarily for due-date objectives.

Table 1: Performance comparison of three dispatching rules in minimizing mean flowtime under conditions of no limit on amount of inserted idle time, no inserted idle time at all, and inserted idle time capped at given value.

	SPT				LWKR				TWPT			
Arrival Rate	No IIT limit	IIT = 0	Limited IIT	Cap	No IIT limit	IIT = 0	Limited IIT	Cap	No IIT limit	IIT=0	Limited IIT	Cap
High	5340.3	1151.0	1149.5	2	4799.1	1330.5	1328.0	2	1411.7	1213.5	1213.4	1
Medium	3262.7	840.1	837.9	**2***	3191.8	932.5	932.0	2	911.5	864.0	863.9	4
Low	1208.7	648.5	646.3	**5***	1588.3	712.8	712.4	2	693.4	669.6	669.0	3

In Tables 2 and 3, similar information is given for the dispatching rule performances for the mean tardiness and number of tardy jobs, respectively. In Table 2, the results for each dispatching rule are given in four rows, giving the average mean tardiness under the conditions of : limited idle time insertion (∞); no inserted idle time allowed (0); best performance under a cap on the amount of inserted time allowed (LIIT); and the value of that cap. The same arrangement is repeated in Table 3 for the average number of tardy jobs. The figures in the columns of Tables 2 and 3 display the results from the different runs combining due-date tightness (Z) and arrival rate (high, medium or low).

4.0 Discussion and Conclusions

The results displayed in Tables 1, 2 and 3 summarize the observed performances for the different dispatching rules in the cases where idle time insertion is allowed, compared to the alternative of no idle times permitted. In each case, a pairwise t-test comparison is done to determine whether an observed difference between the results with and without idle time insertion is statistically significant. Significant differences are indicated by means of an asterisk attached to the value for the maximum allowed idle time that produced the result, which is the value printed under the column headed ‘Cap’. A review of Table 1 shows that there is no statistical significance for any improvements in average mean flowtimes resulting from the use of inserted idle times by the LWKR and TWPT rules. However, the difference for SPT is statistically significant in the medium to low arrival rate cases, although that difference is minor in magnitude.

With regard to the mean tardiness criterion, the only rule to experience some clear advantage from inserted idle time is the COVERT rule, under medium to low due-date tightness levels. Even then, the magnitude of the differences is rather slight. The SPT rule does show statistically significant improvement when inserted idle time is used under low to medium congestion, with very tight due-dates, but this only under a severely limited allowance for inserted idle time.

The results for the number of tardy jobs are markedly different. With the exception of the COVERT and Critical Ratio rules, all the dispatching rules demonstrate statistically significant improvement in performance with inserted idle times, for all arrival levels under tight due-dates (Z=2). The magnitude of this improvement is substantial for many of these dispatching rules, amounting to as much as 26% improvement for the LWKR operating under a high arrival rate and tight due-dates. This improvement is limited for most of the dispatching rules to tight due-date levels (Z=2). As the due-date tightness relaxes, it is only the MDD and MOD rules which continue to exhibit modest improvements with inserted idle times. For very loose due-dates (Z=8), none of the dispatching rules demonstrate any gain by inserting idle times.

The results compiled from this study may be explained to a great extent by the fact that inserted idle time tends to contribute to increased job completion times, as any idle time cannot be recouped later on. The earliest time a job can be completed on a given machine is only extended if idle time is inserted on that machine prior to the job’s arrival. This explains the failure of inserted idle time to have a major impact on the mean flowtime and mean tardiness criteria, the minimization of both of which depends on job completion times.

Table 2: Average mean tardiness by dispatching rules under scenarios of unlimited, none and capped allowances on inserted idle time.

	Z = 2			Z = 4			Z = 6			Z = 8		
Arrival Rate	High	Med.	Low	High	Med.	Low	High	Med.	Low	High	Med.	Low
Covert- ∞*	4217.5	2007.1	526.5	3151.1	772.4	123.3	1803.4	196.1	19.3	846.1	45.8	3.6
Covert -0*	727.2	411.9	228.7	451.3	180.5	61.2	240.0	63.4	12.2	105.3	18.3	3.1
Covert- LIIT*	727.1	411.3	227.8	451.2	180.1	60.7	239.6	63.3	12.1	104.9	18.1	3.0
Cap	1	2	3*	1	2	6*	1*	2*	8*	2*	4*	10*
CR- ∞	1376.6	697.9	367.9	904.1	333.9	114.0	550.6	145.8	30.5	315.1	59.2	9.8
CR- 0	1168.9	608.5	326.9	727.3	248.7	67.4	376.8	72.7	8.0	160.2	12.2	0.5
CR - LIIT	1168.5	608.2	326.2	727.1	248.6	67.0	376.5	72.7	7.9	160.2	12.1	0.5
Cap	2	2	4*	1*	1	8	2	0	8	0	3	0
EDD- ∞	1325.3	683.6	360.3	857.2	305.2	88.2	481.8	104.5	14.0	238.8	25.6	2.0
EDD- 0	1315.7	669.5	344.5	828.4	274.5	70.3	431.7	76.9	12.3	188.3	12.3	0.6
EDD - LIIT	1315.7	669.5	344.3	828.4	274.5	70.3	431.7	76.9	12.1	188.3	12.3	0.6
Cap	0	0	2*	0	0	0	0	0	4	0	0	0
LWK- ∞	4347.6	2747.3	1149.8	4139.0	2560.0	986.6	3993.9	2436.9	889.2	3874.2	2338.0	815.0
LWK- 0	856.3	471.7	264.5	653.7	308.2	139.3	545.4	229.7	88.5	467.4	177.9	59.2
LWK - LIIT	856.3	471.7	264.5	653.7	308.2	139.3	545.4	229.7	88.5	467.4	177.9	59.1
Cap	0	0	0	0	0	0	0	0	0	0	0	1
MDD- ∞	1269.2	565.1	292.3	682.8	234.2	64.9	355.5	75.1	9.7	167.1	17.6	1.5
MDD- 0	874.8	467.3	254.2	531.4	185.7	50.7	269.7	52.1	6.4	110.8	8.1	0.5
MDD - LIIT	874.8	466.8	253.6	531.2	185.7	50.7	269.7	52.1	6.3	110.8	8.1	0.5
Cap	0	2	3	1	0	0	0	0	4	0	0	0
MOD- ∞	3904.4	1794.8	494.7	2230.0	647.5	148.7	1408.5	283.6	51.2	938.4	139.7	14.0
MOD- 0	677.1	374.7	198.1	359.5	138.2	42.8	178.7	48.1	11.5	82.4	16.6	3.2
MOD - LIIT	677.1	373.4	196.0	359.3	138.1	42.4	178.6	47.9	11.4	82.4	16.5	3.1
Cap	0	3	5*	1	1	5	2	2	4*	0	1*	7
SPT- ∞	4874.6	2803.5	759.7	4640.3	2594.7	593.4	4483.8	2464.9	504.4	4355.8	2363.0	441.4
SPT- 0	676.6	382.0	205.4	481.8	229.2	92.1	389.1	166.3	55.1	326.8	127.7	35.7
SPT- LIIT	676.1	380.8	204.5	481.6	229.2	92.0	389.1	166.3	55.1	326.8	127.7	35.7
Cap	2	2*	2*	1	0	1	0	0	0	0	0	0
TWPT- ∞	938.6	452.2	247.5	739.5	293.1	128.5	633.2	219.4	81.8	557.4	172.2	55.8
TWPT- 0	737.7	402.1	221.8	540.5	245.4	105.4	440.8	176.9	63.3	371.3	134.3	40.8
TWPT- LIIT	737.7	401.8	221.6	540.5	245.3	105.4	440.8	176.8	63.3	371.3	134.3	40.7
Cap	0	1	3	0	1	0	0	1	0	0	0	3

*∞ - no limit on inserted idle time; 0 - no idle time insertion allowed; LIIT - idle time insertion allowed up to cap .

The situation is not the same for the minimum number of tardy jobs criterion. As far as that criterion is concerned, the effect of a machine working on a job that is already tardy is equivalent to the machine remaining idle. Increasing the completion time of a job that is already past its due-date does not affect the performance in terms of number of jobs tardy. Hence, there is very good reason for a machine to remain idle, rather than start processing a tardy job, in order to wait for an upcoming job that has a chance for completion before its due-date. Now, in a busy flowshop where many in-process jobs are already overdue, a liberal allowance of inserted idle times will enable machines to avoid working on the already tardy jobs and focus instead on expediting the jobs that can be completed on time. This is seen from the results where job congestion is high and due-dates are tight (Z=2), where the dispatching rules' perform better with high allowances on permissible idle time insertions. Conversely, in the low congestion and loose due-date situations, the number of overdue jobs that are in-process is quite low, so that there is little justification for increasing job completion times by inserting idle times.

Table 3: Average number of tardy jobs by dispatching rules under scenarios of unlimited, none and capped allowances on inserted idle time.

	Z = 2			Z = 4			Z = 6			Z = 8		
Arrival Rate	High	Med.	Low	High	Med.	Low	High	Med.	Low	High	Med.	Low
Covert- ∞*	1990.2	1981.3	1848.5	1922.9	1658.7	923.5	1721.9	920.2	260.7	1309.2	417.9	76.9
Covert - 0*	1952.0	1809.9	1588.4	1726.3	1201.7	706.0	1350.3	641.4	213.0	892.7	285.5	69.0
Covert- LIIT*	1951.5	1808.8	1588.4	1726.3	1201.6	705.9	1349.7	641.3	212.0	892.6	285.4	69.0
Cap	**2**	**1***	**0**	**0**	**1**	**1**	**1**	**2**	**3***	**1**	**1**	**0**
CR- ∞	1988.6	1936.7	1788.7	1883.6	1515.5	1013.2	1648.0	1000.5	435.2	1330.3	607.2	185.5
CR- 0	1981.7	1886.3	1682	1780.1	1214.3	662.4	1321.9	545.5	115.6	751.4	147.3	20.3
CR – LIIT	1981.7	1886.3	1682	1779.2	1213.9	661.1	1321.6	545.5	114.9	750.5	144.9	20.3
Cap	**0**	**0**	**0**	**5**	**3**	**2**	**1**	**0**	**5**	**2**	**3**	**0**
EDD- ∞	1978.5	1882.9	1677.4	1773	1190.5	630.3	1353.5	554.7	102.9	868.2	180.9	18.9
EDD- 0	1976.8	1866.9	1639.9	1734.4	1085.7	517.7	1242.9	439.6	71.8	648.3	97.9	9.15
EDD - LIIT	1976.5	1865.4	1635.5	1734.4	1085.7	517.4	1242.9	439.5	71.4	648.3	97.9	9.15
Cap	**10**	**3***	**7***	**0**	**0**	**1**	**0**	**1**	**3**	**0**	**0**	**0**
LWK- ∞	1057.3	992.9	905.5	535.5	465.3	376.2	408.8	338.3	256.6	346.1	282.6	199.8
LWK- 0	1333.3	1156.5	965.2	434.9	324.7	220	281.2	193.3	114.4	209.2	132.3	66.1
LWK - LIIT	1057.3	992.9	885.3	434.9	323.9	220	281.2	193.3	114.4	209.2	132.3	66.1
Cap	**∞***	**∞***	**16***	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**
MDD- ∞	1837.2	1711.8	1498.5	1585.2	1022.4	521.9	1149.2	451.9	83.8	697.2	139.6	14.8
MDD- 0	1897.3	1735.1	1492.8	1534.7	918.4	433.4	1028.2	354.7	57.4	502.4	74.3	7.2
MDD - LIIT	1849.3	1691.2	1459.2	1525.1	915.2	431.3	1023.9	350.8	56.3	501.4	74.3	7.2
Cap	**∞***	**19***	**20***	**10***	**6**	**8***	**3**	**7***	**4**	**2**	**0**	**0**
MOD- ∞	1795.5	1738.2	1510.2	1545.7	1239.4	634	1372.7	735.7	235.3	1198.7	437	71.2
MOD- 0	1707.8	1473.3	1215	894.6	457.4	191.2	412.5	123.7	31.1	139.5	31.6	7.3
MOD - LIIT	1685.9	1445	1186.2	889.1	450.3	184.2	410.9	121.3	29.9	138.7	31.6	7.3
Cap	**10***	**9***	**10***	**4**	**5**	**6***	**2**	**3***	**5***	**2**	**0**	**0**
SPT- ∞	1279.4	1189.9	1054.5	659.2	551.8	406.3	504.4	405.6	257.9	429.2	332.8	196.8
SPT- 0	1400.5	1207.3	1001.3	440.7	315.2	204.0	271.5	175	91.4	193.3	115.8	52.5
SPT- LIIT	1257.6	1096.1	915.5	438.5	312.35	203.2	270.4	174.3	91.3	192.9	115	52.1
Cap	**19***	**19***	**20***	**2**	**2**	**4**	**2**	**1**	**1**	**1**	**3**	**3**
TWPT- ∞	1272.5	1099.7	907.2	426.35	309.8	203.3	275	175.5	101.5	202.4	117.1	60.6
TWPT- 0	1345.0	1171.1	966.0	403.7	294.2	189.6	253.9	161.9	89.4	182.0	105.0	51.2
TWPT- LIIT	1272.5	1099.7	907.2	403.7	292.9	189.5	253.3	161.6	88.6	181.3	104.8	51.2
Cap	**∞***	**∞***	**∞***	**0**	**9**	**1**	**2**	**1**	**3**	**2**	**1**	**0**

*∞ - no limit on inserted idle time; 0 - no idle time insertion allowed; LIIT - idle time insertion allowed up to cap .

The computational analysis conducted in this research study leads to the following conclusions. First, the use of any amount of inserted idle time in a flowshop is rarely beneficial for minimizing either the mean flowtime or the mean tardiness performance criteria. In those few circumstances where some benefit has been observed, it is a very minor benefit and achieved only with a very strict limit on the permissible amount of idle time. On the other hand, there is strong evidence that inserted idle time helps many dispatching rules to improve their performance in minimizing the number of tardy jobs, particularly when the due-dates are very tight. Second, it is observed that different dispatching rules react differently to the use of inserted idle times. Some rules are able to improve their performance much more substantially than others under the same set of circumstance. Finally, and most critically, there should be a limit on how much idle time can be inserted. This limit depends on the dispatching rule, the shop conditions (congestion, due-date tightness), and the performance criteria. In the absence of such limitation, the performance of most of the dispatching rules rapidly deteriorates under most of the operating conditions considered in this study.

The objective of the research conducted in this study has been to determine the effect of inserted idle times in the performance of some well-known dispatching rules in minimizing a selection of performance criteria in a dynamic flowshop that receives jobs continuously over time. The main finding is that inserted idle time can provide opportunity to improve the performance of these dispatching rules in minimizing the number of tardy jobs. What remains to be seen is whether such improvement is enough to make these dispatching rules stronger competitors in comparison to other scheduling algorithms specifically designed to minimize the number of tardy jobs, such as those proposed by Rajendran and Holthaus (1999), Lodree (2004) and Chiang and Fu (2007). Futhermore, investigation needs to be also conducted on those procedures themselves to determine whether inserted idle times may provide opportunity for further improvements in their performances.

Future research efforts should also investigate the location and duration of inserted idle times. This means that in addition to deciding which job to load next, a dispatching rule will also need to decide how much idle time, if any, to allow in waiting for an upstream job to arrive. Finally, Inserted idle times have been also the subject of research for non-regular performance measures, such as such as sum of earliness/tardiness penalties. Future research may also consider inserted idle time in such non-regular performance measures for dynamic flowshops similar to the one discussed in this paper.

References

Baker, K.R, *Introduction to Sequencing and Scheduling,* John Wiley & Sons, Inc., New York, 1974.

Baker, K.R. and Bertrand, J.W.M., Comparison of due-date selection rules, *AIIE Transactions (American Institute of Industrial Engineers),* vol. 13, no. 2, pp. 123-131, 1981.

Baker, K.R. and Kanet, J.J. 1983, "Job shop scheduling with modified due dates, *Journal of Operations Management,* vol. 4, no. 1, pp. 11-22.

Blackstone Jr., J.H., Phillips, D.T. and Hogg, G.L., State-of-the-art survey of dispatching rules for manufacturing job shop operations, *International Journal of Production Research,* vol. 20, no. 1, pp. 27-45, 1982.

Carroll, D.C., Heuristic sequencing of jobs with single and multiple components, *Ph.D. Dissertation,* Sloan School of Management, MIT, Cambridge, MA, 1965.

Chiang, T.C. and Fu, L.C., Using dispatching rules for job shop scheduling with due date-based objectives, *International Journal of Production Research,* vol. 45, no. 14, pp. 3245-3262, 2007.

Conway, R.W., Maxwell, W.L. and Miller, L.W., *Theory of Scheduling,* Addison-Wesley, Reading, MA, 1967.

El-Bouri, A., A cooperative dispatching approach for minimizing mean tardiness in a dynamic flowshop, *Computers and Operations Research,* vol. 39, no. 7, pp. 1305-1314, 2012.

Giffler, B. and Thompson, G.L., Algorithms for solving productin scheduling problems, *Operations research,* vol. 8, no. 4, pp. 487-503, 1960.

Haupt, R., A survey of priority rule-based scheduling, *OR Spektrum,* vol. 11, no. 1, pp. 3-16, 1989.

He, W. and Sun, D., Scheduling flexible job shop problem subject to machine breakdown with route changing and right-shift strategies, *International Journal of Advanced Manufacturing Technology,* vol. 66, no. 1-4, pp. 501-514, 2013.

Kanet, J.J. and Sridharan, V., Scheduling with inserted idle time: problem taxonomy and literature review, *Operations research,* vol. 48, no. 1, pp. 99-110, 2000.

Kim, Y., Heuristics for flowshop scheduling problems minimizing mean tardiness, *Journal of the Operational Research Society,* vol. 44, no. 1, pp. 19-28, 1993.

Lodree Jr., E., Jang, W., and Klein, C.M., A new rule for minimizing the number of tardy jobs in dynamic flowshops, *European Journal of Operational Research,* vol. 159, no. 1, pp. 258-263, 2004.

Luo, H., Zhang, A. and Huang, G.Q., Active scheduling for hybrid flowshop with family setup time and inconsistent family formation, *Journal of Intelligent Manufacturing,* , in press, 2013.

Morton, T.E. and Pentico, D.W., *Heuristic Scheduling Systems,* John Wiley & Sons, Inc., New York, 1993.

Morton, T.E. and Ramnath, P., *Guided Forward TABU/Beam Search for Scheduling Very Large Dynamic Job Shops,* Working Paper, Carnegie Mellon University, Pittsburgh, PA, 1992.

Pinedo, M., *Scheduling Theory, Algorithms, and Systems,* second edition, Prentice-Hall, New Jersey, 2002.
Rajendran, C., Heuristic algorithm for scheduling in a flowshop to minimize total flowtime, *International Journal of Production Economics*, vol. 29, no. 1, pp. 65-73, 1993.
Rajendran, C. and Holthaus, O., Comparative study of dispatching rules in dynamic flowshops and jobshops, *European Journal of Operational Research,* vol. 116, no. 1, pp. 156-170, 1999.
Soroush, H.M., Solving a stochastic single machine problem with initial idle time and quadratic objective, *Computers and Operations Research*, vol. 37, no. 7, pp. 1328-1347, 2010.
Storer, R.H. and Avci, S., Algorithms for non-regular job shop scheduling problems, *IIE Annual Conference and Exhibition 2004*, pp. 199, Houston, TX, USA, May 14-19, 2004.
Yoon, S.-. and Ventura, J.A., Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling, *Computers and Operations Research,* vol. 29, no. 10, pp. 1301-1315, 2002.
Valente, J.M.S and Schaller, J.E., Improved heuristics for the single machine scheduling problem with linear early and quadratic tardy penalties, *European Journal of Industrial Engineering*, vol. 4, no. 1, pp. 99-129, 2009.