International Journal of Applied Science and Engineering
2006.4, 3: 259-273

Comparing Steady-state Performance of Dispatching Rule-pairs in Open Shops

Horng-Chyi Horng *

*Department of Industrial Engineering and Management,
Chaoyang University of Technology, Wufong, Taichung 41349, Taiwan, R.O.C.*

Abstract: Unlike implemented in other production systems, dispatching in an open shop not only takes into account the job priority at each machine but also the selection of the next machine for any jobs leaving the current machine. The control mechanism for scheduling in an open shop can be utilized by a dispatching rule-pair that consists of a machine-selection rule and a job-dispatching rule. This study conducts a steady-state simulation comparison on the performance of 39 dispatching rule-pairs in open shops. We find that using NINQ as the machine-selection rule can minimize the mean flowtime of jobs while using LINQ as the machine-selection rule can minimize mean tardiness of jobs for most cases. The choice of the best dispatching rule-pair depends on the selected performance criterion as well as the system's configurations such as utilization factor, number of machines, and processing time distribution of jobs. Finally, under similar system configurations, the best job-dispatching rule in an open shop is different from that of a job shop.

Keywords: Open shops; Steady-state Simulation; Dispatching Rules.

Introduction

Dispatching rule is the most simplified and easy-to-implement scheduling tool for a production system. Over the past few decades, many dispatching rules have been developed and proven to be effective for flow shops and job shops. As far as flow shops are considered, Hunsucker and Shah[1] compared six job-dispatching rules for a constrained flow shop with multiple processors and found that SPT rule should be adopted under makespan($C_{\max}$) and mean flowtime($\overline{F}$) criteria. In addition, they found that using FIFO (first in first out) rule minimizes $F_{\max}$ when the congestion level of the system is very high. While considering dynamic job arrival in flow shops, Sarper and Henry[2] showed that MDD (modified due date) and SPTL (shortest processing time local) perform better than others in $\overline{F}$, $F_{\max}$, and mean tardiness ($\overline{T}$) criteria for different utilization rates and due date factors. In dynamic flow shops, Barrett and Kadipasaoglu[3] evaluated the performance of five dynamic and four static dispatching rules. They found that SPT performs well in $\overline{F}$ and percentage of tardiness ($\%T$), but worse in $\overline{T}$. They also found that there is no evidence that the performance differences between dynamic dispatching rules and static dispatching rules are significant. Rajendran and Holthaus[4] generated new rules by

* Corresponding author; e-mail: hchorng@cyut.edu.tw
Accepted for Publication: Novemberr 18, 2006

compounding processing time with other information such as work in the next queue and job arrival time. They showed that, based on $\overline{F}$ criterion, PT + WINQ (processing time plus work in next queue) performs as good as SPT rule. Table 1 is a list of best dispatching rules summarized from literature for flow shops under different performance criteria.

Table 1. Summary of the best dispatching rules for flow shops

Criteria	Best Rule(s)	References
$\overline{F}$	SPT	Hunsucker and Shah(1994)
	MDD、SPTL	Sarper and Henry(1996)
	SPT	Barrett and Kadipasaoglu(1990)
	SPT、PT+WINQ	Rajendran and Holthaus(1999)
$F_{max.}$	SPT、FIFO	Hunsucker and Shah(1994)
	MDD、SPTL	Sarper and Henry(1996)
	FIFO、AT、AT-RPT、PT/TIS、(PT+WINQ)/TIS、PT+WINQ+AT	Rajendran and Holthaus(1999)
σ_F^2	PT/TIS、(PT+WINQ)/TIS	Rajendran and Holthaus(1999)
$\%T$	SPT	Barrett and Kadipasaoglu(1990)
	RR、SPT	Rajendran and Holthaus(1999)
$\overline{T}$	MDD、SPTL	Sarper and Henry(1996)
	CR、OPCR	Barrett-Kadipasaoglu(1990)
	RR、COVERT	Rajendran and Holthaus(1999)
$T_{max.}$	RR、PT+WINQ+SL	Rajendran and Holthaus(1999)
σ_T^2	RR、PT+WINQ+SL	Rajendran and Holthaus(1999)

Ramasesh[5] reviewed simulation-based research on dynamic job shop scheduling from the year 1965 to 1986 and found that SPT is the best rule for minimizing $\overline{F}$ but has higher variance of flowtime (σ_F^2) than others. For job shops with low, medium, and high utilization rate, Waikar *et al.*[6] evaluated 10 dispatching rules. They showed that SPT and LWKR (least work remaining) minimize $\overline{F}$ and the percentage of tardiness of jobs, while DDT (due date time) and SLK/RO (least ratio of slack time remaining to number of operations remaining) perform well under $\overline{T}$ criterion. Holthaus[7] evaluated compounded rules in job shops and showed that PT + WINQ, AT-RPT (arrival time minus remaining processing time), SPT, RR (rule created by Raghu and Rajendran[8]), and PT + WINQ + SL (processing time plus work in next queue plus slack) perform well in $\overline{F}$, F_{max}, σ_F^2, $\%T$, $\overline{T}$, maximum tardiness(T_{max}), and tardiness variance criteria(σ_T^2), respectively. Another rule created by Holthaus and Rajendran[9], PT+WINQ+AT+SL (processing time plus work in next queue plus arrival time plus slack), also performs well in F_{max}, σ_F^2, T_{max}, and σ_T^2 criteria. They further compared flow shops and job shops, and showed that different production type and job routings have significant impact on the selection of the best dispatching rule over different performance criteria[4]. In an attempt to improve job shop performance, they further developed new rules and found that 2PT+WINQ+NPT (double processing time plus work in next queue plus processing time at next operation) is better than PT+WINQ in minimizing $\overline{F}$ while PT+WINQ+NPT+WSL (processing time plus work in next queue plus processing time at next operation plus "WINQ" slack) outperforms PT+WINQ+SL in minimizing $T_{\max}$ and σ_T^2 [10]. Considering mean time to failure and repair time of machines in the job shop, Liu [11] found that FIFO should be adopted for minimizing the $\overline{T}$ and the $\%T$ of jobs. Table 2 lists the best dispatching rules for job shops summarized from literature under different performance criteria.

An open shop is often considered as a special case of a job shop where jobs can be routed in any way through the system as long as all necessary operations are performed. Examples of open shops are maintenance and repair of automobiles, health examination of human body, and quality inspection of products. The scheduling problem in an open shop is more complicated than others since jobs do not follow a certain route through the system. Therefore, analytic methods can only be applied to analyze simple open shop problem (less than 3 stations)[12]. Unlike implemented in other production systems, scheduling in an open shop not only takes into account job priority at each machine but also the selection of the next machine for any jobs leaving the current machine. The control mechanism for scheduling in an open shop can be utilized by a dispatching rule-pair that consists of a machine-selection rule and a job-dispatching rule. Therefore, the purpose of this study is to perform a steady-state simulation study of open shops to investigate the performance of 39 dispatching rule-pairs under several standard performance measures.

In the following section, a description of various job-dispatching rules as well as machine-selection rules is given. Section 3 and 4 discuss the design of the simulation experiments and the analysis of simulation results, respectively. The last section presents the conclusions of this study and suggestions for future studies.

2. Dispatching rules

As mentioned above, the open shop considered in this study requires that all jobs be processed by every machine in the shop and the routing of each job is immaterial. That is, for a three-machine open shop, the possible routings for any job are M_1-M_2-M_3, M_1-M_3-M_2, M_2-M_1-M_3, M_2-M_3-M_1, M_3-M_1-M_2, and M_3-M_2-M_1. Therefore, each dispatching rule-pair in an open shop consists of one machine-selection rule and one job-dispatching rule. Once a job's operation at the current machine is completed, a machine-selection rule is applied to select the next machine for that job. On the other hand, when a machine has completed the current job, a job-dispatching rule is applied to decide which job in the queue of the machine is to be processed next. Since there are a lot of researches focusing on dispatching rules in job shops[13-15], this study will inherit their job-dispatching rules except those cannot be implemented in open shops.

2.1 Machine-Selection Rules

Three most commonly used machine-selection rules are considered in this study. Let U be the set that contains all machines that a given job has not yet visited, these three machine-selection rules are defined as below.

a. **Random:** Each machine that a given job has not yet visited has equal probability being selected as the next machine. That is, $P(M_i) = P(M_j)$, for all $M_i, M_j \in U$.
b. **Number in Next Queue (NINQ):** Taking all not yet visited machines for a give job into consideration, the machine with the smallest number of jobs waiting is selected as the next machine. That is, select the machine with Min.$\{NQ(M_i)\}$, $M_i \in U$,

where $NQ(M_i)$ is the number of jobs waiting in the queue of machine M_i.

c. **Length in Next Queue (LINQ):** Considering all not yet visited machines for a give job, the machine with the shortest total expected processing time of jobs in the queue is selected as the next machine. That is, select machine with $Min.\{LQ(M_i)\}$, $M_i \in U$, where $LQ(M_i) = \Sigma PT_j$ and PT_j is the expected process time of the j^{th} job in the queue of machine M_i.

2.2 Job dispatching rules

Thirteen job-dispatching rules are considered in this study. Most of them have been studied for job shops with different system configurations. All rules discussed below select the job with the minimum value of Z_{ij} as the next job to be processed on the machine, where Z_{ij} is the priority index and is defined differently for each rule.

a. **First In First Out (FIFO):** This rule selects the next job from the queue based on their arrival time at the current machine. That is, $Z_{ij} = r_{ij}$, where r_{ij} is the arrival time of job i at machine M_j.

b. **Arrival Time (AT):** This rule selects the next job from the queue based on their arrival time into the system. That is, $Z_{ij} = r_i$, where r_i is the arrival time of job i into the system.

c. **Arrival Time-Total Remaining Processing Time (AT-RPT):** This rule, provided by Rajendran and Holthaus [4], selects the next job from the queue based on their arrival time into the system with respect to the total remaining processing time. The formula is, $Z_{ij} = -(t - r_i) - RPT_i$, where t is the current time and RPT_i is the total remaining process time of job i.

d. **Shortest Processing Time (SPT):** This rule selects the next job from the queue based on their processing times at the current machine. That is, $Z_{ij} = p_{ij}$, where p_{ij} is the processing time of job i at machine M_j.

e. **Earliest Due Date (EDD)**: This rule selects the next job from the queue based on their due date. That is, $Z_{ij} = d_i$, where d_i is the due date of job i.

f. **Minimum Slack Time (MST):** This rule selects the next job from the queue based on their slack times. Slack time of any job is computed by deducting the current time and the total remaining process time from the due date of the job. That is, $Z_{ij} = s_i = d_i - RPT_i - t$.

g. **Modified Due Date (MDD):** This rule selects the next job from the queue based on their due dates with respect to the current time and the total remaining processing time of the job. That is, $Z_{ij} = Max\{d_i, t + RPT_i\}$.

h. **Critical Ratio (CR):** This rule selects the next job from the queue based on their relatively available time divided by the total remaining process time of the job. That is, $Z_{ij} = (d_i - t) / RPT_i$.

i. **Slack per Remaining Operation (S/PMOP):** This rule selects the next job from the queue based on their slack time divided by the number of remaining operations of the job. That is, $Z_{ij} = (d_i - RPT_i - t) / RO_i$, where RO_i is the total remaining operations of job i.

j. **RR:** This rule, provided by Raghu and Rajendran[8], can improve the average delay time and the average flowtime performance. The formula takes into account the total work content of job i and is represented by

$$Z_{ij} = \left(s_i \times \exp(-u) \times p_{ij}\right) / RPT_i + \exp(u) \times p_{ij} + \overline{w_i}$$

, where u is the machine utilization. In job shops, $\overline{w_i}$ in the formula for rules RR, PT+WINQ, PT+WINQ+AT, and PT+WINQ+SL represents the total work content of job i, that is, the average waiting time for job i at the next unvisited machine. However, it must be modified to represent the average waiting time of all remaining unvisited machines for job i when applied to open shops.

k. **Processing Time + Work in Next Queue (PT+WINQ):** This rule, provided by Holthaus and Rajendran [9], can improve the average flowtime. The formula is $Z_{ij} = p_{ij} + \overline{w_i}$.

l. **Processing Time + Work in Next Queue + Arrival Time (PT+WINQ+AT):** This rule, provided by Holthaus and Rajendran[9], can improve the maximum flowtime and the flowtime variance. The formula is $Z_{ij} = p_{ij} + \overline{w_i} + r_i$.

m. **Processing Time + Work in Next Queue + Negative Slack (PT+WINQ+SL):** This rule, provided by Holthaus and Rajendran[9], can improve the maximum delay time and its variance. The formula is $Z_{ij} = p_{ij} + \overline{w_i} + s_i$.

Table 2. Summary of the best dispatching rules for job shops

Performance Criteria	Best Rule(s)	References
$\overline{F}$	SPT、LWKR	Waikar *et al.*(1995)
	PT+WINQ	Holthaus(1997)
	PT+WINQ	Holthaus and Rajendran(1997)
	RR、SPT、PT+WINQ	Rajendran and Holthaus(1999)
	2PT+WINQ+NPT	Holthaus and Rajendran(2000)
$F_{max.}$	AT-RPT	Holthaus(1997)
	PT+WINQ+AT、PT+WINQ+AT+SL	Holthaus and Rajendran(1997)
	AT-RPT、PT+WINQ+AT	Rajendran and Holthaus(1999)
σ_F^2	AT-RPT	Holthaus(1997)
	PT+WINQ+AT、PT+WINQ+AT+SL	Holthaus and Rajendran(1997)
	AT-RPT、PT+WINQ+AT	Rajendran and Holthaus(1999)
$\%T$	MOD	Baker and Kanet(1983)
	SPT、LWKR	Waikar *et al.*(1995)
	SPT	Holthaus(1997)
	SPT	Holthaus and Rajendran(1997)
	FIFO	Liu(1998)
	RR、SPT	Rajendran and Holthaus(1999)
	2PT+WINQ+NPT	Holthaus and Rajendran(2000)
$\overline{T}$	MOD	Baker and Kanet(1983)
	DDT、SLK/RO	Waikar *et al.*(1995)
	RR	Holthaus(1997)
	RR	Holthaus and Rajendran(1997)
	FIFO	Liu(1998)
	RR	Rajendran and Holthaus(1999)

Table 2. Summary of the best dispatching rules for job shops(continued)

$T_{max.}$	RR、PT+WINQ+SL	Holthaus(1997)
	PT+WINQ+SL、PT+WINQ+AT+SL	Holthaus and Rajendran(1997)
	RR、PT+WINQ+SL	Rajendran and Holthaus(1999)
	PT+WINQ+NPT+WSL	Holthaus and Rajendran(2000)
σ_T^2	RR、PT+WINQ+SL	Holthaus(1997)
	PT+WINQ+SL、PT+WINQ+AT+SL	Holthaus and Rajendran(1997)
	RR、PT+WINQ+SL	Rajendran and Holthaus(1999)
	PT+WINQ+NPT+WSL	Holthaus and Rajendran(2000)

3. Design of simulation experiments

This study is intended to conduct steady-state simulation experiments to get solid insight about dispatching rule-pairs in open shops. Therefore, three problems to be considered are:

a. What is the best machine-selection rule in general for open shops?
b. What are the best dispatching rule-pairs under a set of performance criteria for open shops?
c. Is the best job-dispatching rule for open shops different from that of job shops given similar system configurations?

In the following sections, the assumptions for the open shop system considered in this study as well as designs of experiments corresponding to three problems mentioned above are briefly discussed.

3.1 Assumptions

Several assumptions are made to further specify the open shop system considered in this study. Firstly, there are *n* independent jobs and *m* irrelevant machines. Secondly, each job must be processed by every machine in the system and each machine can only process one job at a time. In addition, job-preemption, maintenance, and repair are not considered. Each job is also unique, that is, there is no priority or assembly relationship between jobs. Finally, the setup time for each job is ignored.

3.2 System Descriptions and Model Validation

As most research in this field did, this study uses exponential distributions to model the interarrival times of jobs. The performance criteria considered are mean flowtime($\overline{F}$), maximum flowtime(F_{max}), variance of flowtime(σ_F^2), proportion of tardy jobs(*%T*), mean tardiness($\overline{T}$), maximum tardiness(T_{max}), and variance of tardiness(σ_T^2). Each simulation will be terminated when 25,000 jobs are completed. Statistical Data are truncated and reset at 16,000 time units when the system reaches steady state. Twenty simulation replicates are made for each system configuration. It is sufficient to test differences in performance based on a procedure described by Law and Kelton[16]. The simulation model was written in ARENA 5.0[17] language and verified by performing the following two steps. Firstly, detailed traces during simulation runs of the model were generated. It showed that the simulation models performed properly. Secondly, operational statistics such as machine utilization collected from running the simulation model were equal to their pre-setting values. Furthermore, the simulation model is validated via the comparison of the average completion time of all jobs to that of the published paper[18] and calculated lower bound (LB). The comparison is listed in Table 3, where the lower bound of the average completion time of all jobs, also known as the average total weighted completion time, is defined as $\sum_{i=1}^{n}\sum_{j=1}^{m} w_i p_{ij}$. The simulation results of this study and that of the published paper are not statistically significant for systems with small job/machine numbers, but the former is better for systems with large job/machine numbers. The lower bound limitations for all cases are satisfied.

3.3 Due-date Setting

Due-date setting will affect tardiness related performance criteria. Like most research, this study adopts the total works method (TWK) in calculating due-date for jobs. Based on overall information or individual information, this method is further categorized into job due-date based and operation due-date based. The due-date calculations are

Job due-date setting: $d_i = r_i + c\sum_{j=1}^{m} p_{ij}$ and

Operation due-date setting:

$$d_{ij} = r_i + c\sum_{j\in\theta} p_{ij}$$

where θ is the set of processed machines and c is the allowance factor. Most research uses an allowance factor between 2 to 8.

Table 3. Model validation

Model Size		5×5*	10×10	20×20	30×30
Lower Bound		1669	7217	29514	64140
Published Paper	WSPT	2621	11757	53817	122317
	WSTR	2677	13231	57714	128212
	WLTR	2858	13329	58732	128390
	CR+SPT	2776	13370	61722	142876
This Study	WSPT	2515	10865	50233	118740
	WSTR	2632	11727	54741	128170
	WLTR	2664	11322	52697	124320
	CR+SPT	2625	11514	52204	123030

* 5×5: Number of jobs × number of machines in the model.

3.4 Experimental Design

Table 4 lists the experimental designs for the problems considered in this study. Factors considered include machine-selection rules, job-dispatching rules, number of machines, utilization factor, allowance factor, and the distribution types of job processing time. There are a total of 936 experiment combinations and 18,720 simulation runs. Both job inter-arrival time and machine processing time are assumed to be in minutes. The utilization rate of the system is achieved by manipulating the inter-arrival time of jobs and the processing time at each machine. A 95% utilization rate is used to represent a system with high congestion level and an 80% utilization rate for a system with low-to-medium congestion level. The number of machines in the open shop is set to be 5 or 10 to represent small size or medium-to-large size system, respectively.

Table 4. Factor levels for the experiments

Factors	*Levels*
Machine-selection Rules	**Random, LINQ, NINQ**
Job-dispatching Rules	FIFO, AT, AT-RPT, SPT, EDD, MST, MDD, CR, S/RMOP, RR, PT+WINQ, PT+WINQ+AT, PT+WINQ+SL
Number of Machines	5, 10
Utilization Rate	80%, 95%
Allowance Factor	4, 6
Process Time Distributions	Uniform(1,49), EXPO(25), Normal(25,5)

4. Simulation results and analysis

A Portion of the simulation output summary is illustrated in Table 5. Data under each performance criterion are average value calculated from 20 simulation replica. All factors but allowance factor considered in this study are significant (p-value < 0.001) no matter which performance criterion is chosen. Using different levels of allowance factor, 4 or 6, only affects the tardiness related performance criteria. In the following sections, simulation results will be explained according to three problems listed in section 3.

4.1 Machine-selection Rules

As shown in Table 5 for all 13 job-dispatching rules, using *Random* as the machine-selection rule produces the worst results no matter which performance criterion is chosen. We have applied the Fisher's Least Significant Difference (LSD) method for multiple comparisons among three machine-selection rules. The mean flowtime of jobs when using *Random* as the machine-selection rule is about 50% more than that of using NINQ or LINQ for open shop with uniformly distributed process times and low utilization rate. Worsen than that, the mean tardiness of jobs is at least 4 times more. Therefore, we suggest that when dealing with dispatching problem of an open shop, be sure to use a machine-selection rule other than *Random*. We also found that there is no significant performance difference between NINQ and LINQ. However, other things being equal, we found that NINQ produces the minimum mean flowtime of jobs, while LINQ produces minimum mean tardiness of jobs for most cases.

4.2 Dispatching Rule-pairs

This research further applies LSD method for multiple comparisons among all dispatching rule-pairs. Table 6 is the test results with top three dispatching rule-pairs listed for each system configuration and performance criterion. Best dispatching rule-pairs for different system configurations according to each performance criterion are discussed in detail in the following sections.

- Mean Flowtime

  When the process times of jobs on each machine are uniformly or exponentially distributed, SPT related dispatching rule-pairs such as NINQ+SPT and LINQ+SPT outperform others on $\overline{F}$ criterion. Other good choices include NINQ+MDD and NINQ+EDD for uniformly distributed process times, and NINQ+RR and NINQ+PT+WINQ for exponentially distributed process time. For the normally distributed process times, NINQ+MDD, NINQ+EDD, and NINQ+PT+WINQ+AT are top three dispatching rule-pairs to minimize mean flowtime of jobs. In this case, using NINQ+SPT or LINQ+SPT is still a good choice, but the mean flowtime of jobs is about 2.5% more for systems with low utilization rate and 10% more for systems with high utilization rates, respectively. Further analysis reveals that the mean flowtime of jobs for systems using the worst dispatching rule-pair is 55% to 289% more than that of using the best dispatching rule-pair as shown in Figure 1. If excluding *Random* machine-selection rules, the mean flowtime of jobs for systems using the worst dispatching rule-pair is about 8% to 75% more than that of using the best dispatching rule-pair as shown in Figure 2.

- Maximum Flowtime

  NINQ+AT-RPT and LINQ+AT-RPT outperform others in every situation as what founded in job shop literatures. Other dispatching rule-pairs that perform better in this criterion are all arrival time related dispatching rule-pairs such as NINQ+AT, NINQ+MST, and LINQ+FIFO. The performance difference between systems using the best dispatching rule-pair and the worst dispatching rule-pair ranges from 4 to 20 times in maximum flowtime.

- Flowtime Variance

  The flowtime variance of jobs is highly correlated to the maximum flowtime of jobs. Therefore, NINQ+AT-RPT and LINQ+AT-RPT minimize the flowtime variance of jobs in the system for most cases. However, for exponentially distributed process times and low utilization rate, LINQ+PT+WINQ+AT and NINQ+PT+WINQ+AT are the best dispatching rule-pairs. It is worth to mention that NINQ+AT appears to be on the top three for most cases in this criterion. The performance difference between systems using the best and the worst dispatching rule-pair ranges from 6 to 70 times in flowtime variance.

- Percentage of Tardy Jobs

  When the system's utilization rate is low, NINQ+MDD, LINQ+EDD, LINQ+MST, and NINQ+MST are dispatching rule-pairs that can minimize the percentage of tardy jobs. On the other hand, when the system's utilization rate is high, LINQ+SPT and NINQ+SPT become the best dispatching rule-pairs. NINQ+RR is another choice for minimizing the percentage of tardy jobs for exponentially distributed process times. The performance difference between systems using the best and the worst dispatching rule-pair is well over 4 times in percentage of tardy jobs.

- Mean Tardiness

  MDD, EDD, and MST coupled with NINQ or LINQ create dispatching rule-pairs that can minimize the mean tardiness of jobs. These three job-dispatching rules are due-date related. Therefore, it is no surprise that they perform better in this category. LINQ+SPT and NINQ+SPT do not perform well except for systems with exponentially distributed process times and high utilization rate. In this highly congested system and broad ranges of process times, SPT job-dispatching can move most of the jobs out of the system quickly to avoid further job congestion in the system. The performance difference between systems using the best and the worst dispatching rule-pair is at least 8 times in mean tardiness.

■ Maximum Tardiness

As far as the maximum tardiness of jobs is concerned, LINQ+MST and NINQ+MST dominate in all cases but the case when there are 10 machines in the open shop with 80% utilization rate and exponentially distributed process times. In that case, NINQ coupled with EDD or MDD will result in the best outcome for the maximum tardiness criterion. The performance difference between systems using the best and the worst dispatching rule-pair is at least 12 times in maximum tardiness.

■ Tardiness Variance

The same as the mean tardiness criterion, MDD, EDD, and MST coupled with NINQ or LINQ create dispatching rule-pairs that also minimize the tardiness variance of jobs. The only exception is that when the system's utilization is high and the process times are uniformly or exponentially distributed. In that situation, LINQ+CR is the best choice. The tardiness variance of jobs for systems using worst dispatching rule-pair is at least 34 times than that of systems using the best dispatching rule-pair.

Table 5. A portion of the simulation output summary report

System	Job Rule	Machine Rule	$\overline{F}$	F_{max}	σ_F^2	$\%T$	$\overline{T}$	T_{max}	σ_T^2	Avg. WIP
5 80% 4 Uniform	FIFO	Random	390.83	1076.7	27713	25.76	32.911	686.15	7385.9	12.493
		NINQ	256.09	689.98	10793	6.002	3.767	324.72	645.12	8.1572
		LINQ	255.64	666.41	10576	5.734	3.4672	315.82	550.59	8.1439
	AT	Random	389.57	904.98	21323	25.174	29.054	554.72	5181	12.469
		NINQ	244.88	650.72	8241.6	4.626	2.6897	301.05	380.81	7.7997
		LINQ	245.21	660.03	8230.2	4.788	2.805	309.42	417.23	7.81
	AT-RPT	Random	382.18	843.29	19107	24.914	27.758	537.97	4973.5	12.203
		NINQ	249.62	599.84	8103	5.712	3.4586	329.36	507.01	7.9505
		LINQ	249.47	607.82	8124.9	5.748	3.6518	326.66	524.4	7.9462
	SPT	Random	310.01	3179.7	64968	8.39	25.189	2589.8	22993	9.967
		NINQ	238.64	1938.6	18969	2.758	5.4887	1353.7	3860	7.5961
		LINQ	238.67	1881.9	17232	2.61	4.8965	1315.5	3484.3	7.5937
	EDD	Random	373.25	1088.1	32616	17.598	18.405	417.54	3339.3	12.019
		NINQ	239.02	823.87	10661	1.264	0.47277	126.53	54.973	7.6118
		LINQ	239.82	782.94	10318	1.322	0.50937	117.43	56.688	7.638
	MST	Random	367.44	1022.4	27459	14.948	12.786	323.1	1951.3	11.813
		NINQ	241.09	755.33	10723	1.246	0.51116	80.997	60.696	7.6787
		LINQ	242.32	755.06	10497	1.278	0.53673	89.144	63.524	7.7177
	MDD	Random	363.05	1345.2	31920	14.968	14.561	681.25	3374.6	11.62
		NINQ	238.94	855.48	10687	1.214	0.51172	173.17	96.433	7.6094
		LINQ	239.78	832.3	10352	1.242	0.4686	165.93	69.914	7.6364
	CR	Random	385.16	1027.2	25955	25.172	17.986	438.09	2522.5	12.324
		NINQ	269.48	818.2	15176	7.528	2.6543	224.75	245.97	8.584
		LINQ	274.23	854.72	16213	8.888	3.4798	285.45	342.67	8.7358
	S/RMOP	Random	362.58	973.56	27954	15.354	11.12	344.81	1531.9	11.62
		NINQ	257.97	817.6	14122	2.858	0.80925	116.59	77.18	8.2169
		LINQ	260.46	795.84	14103	2.964	0.81061	109.5	74.111	8.2952
	RR	Random	315.5	2763.6	61593	9.422	26.686	2153.7	21618	10.078
		NINQ	247.33	2083.8	23513	3.446	6.6616	1426.1	4448.9	7.8755
		LINQ	248.63	2006.3	21845	3.3	5.6663	1328.5	3562.8	7.9192
	PT+ WINQ	Random	320.14	2573.4	50869	12.29	28.068	2078.4	17829	10.231
		NINQ	241.97	1847.3	18065	4.092	6.6306	1327.9	3966.4	7.7058
		LINQ	243.06	1878.5	17308	4.218	6.856	1348	4381.1	7.7403

Table 5. A portion of the simulation output summary report (comtinued)

	PT+	Random	363.16	852.92	17944	20.4	19.237	465.13	3094.6	11.618
	WINQ+	NINQ	242.73	705.42	8309.1	4.202	2.4471	288.09	361.07	7.7304
	AT	LINQ	243.16	704.26	8211.5	4.338	2.5099	297.81	372.41	7.7444
	PT+	Random	330.92	1488.2	37486	16.528	23.552	950.03	6701.5	10.625
	WINQ	NINQ	241.45	1040.2	12927	4.88	3.6818	470.36	701.84	7.6907
	+SL	LINQ	242.31	992.95	12174	4.996	3.5206	458.04	594.9	7.7194

Table 6. Summary of the best rule-pairs for open shops

System				$\overline{F}$	$F_{\max}$	σ_F^2	$\% T$	$\overline{T}$	$T_{\max}$	σ_T^2
5	80%	4	Uniform	N+SPT*	N+AT-RPT	N+AT-RPT	N+MDD	L+MDD	N+MST	N+EDD
				L+SPT**	L+AT-RPT	L+AT-RPT	L+MDD	N+EDD	L+MST	L+EDD
				N+MDD	N+AT	L+PT+WINQ+AT	N+MST	L+EDD	L+S/RMOP	N+MST
			Expo	N+SPT	L+AT-RPT	L+PT+WINQ+AT	N+SPT	L+MDD	N+MST	L+MDD
				L+SPT	N+AT-RPT	N+PT+WINQ+AT	N+RR	N+MDD	L+MST	L+EDD
				N+RR	L+AT	L+AT	N+MDD	N+EDD	N+S/RMOP	N+EDD
			Normal	N+MDD	N+AT-RPT	N+AT-RPT	L+MST	L+MST	L+MST	L+MST
				N+EDD	L+AT-RPT	L+AT-RPT	N+MST	N+MST	N+MST	N+MST
				N+PT+WINQ +AT	L+MST	N+AT	N+MDD	L+EDD	N+EDD	L+EDD
5	80%	6	Uniform	N+SPT	N+AT-RPT	N+AT-RPT	L+MST	L+MST	L+MST	L+MST
				L+SPT	L+AT-RPT	L+AT-RPT	N+MST	N+MST	L+EDD	N+MST
				N+EDD	N+AT	L+PT+WINQ+AT	L+EDD	L+EDD	L+MDD	L+EDD
			Expo	N+SPT	L+AT-RPT	L+PT+WINQ+AT	N+MDD	N+MDD	N+MST	N+MDD
				L+SPT	N+AT-RPT	N+PT+WINQ+AT	N+MST	N+EDD	L+MST	N+EDD
				N+RR	L+AT	L+AT	N+EDD	N+MST	N+EDD	N+MST
			Normal	N+EDD	N+AT-RPT	N+AT-RPT	N+MST	L+MST	N+EDD	
				N+MDD	L+AT-RPT	L+AT-RPT	L+MST	N+MST	N+MDD	
				N+PT+WINQ +AT	L+AT	N+AT	N+AT-RPT	N+PT+WINQ +AT		
5	95%	4	Uniform	N+SPT	L+AT-RPT	L+AT-RPT	L+SPT	L+MDD	N+MST	L+CR
				L+SPT	N+AT-RPT	N+AT-RPT	N+SPT	N+MDD	L+MST	N+CR
				L+MDD	L+AT	L+AT	L+RR	N+CR	N+EDD	L+EDD
			Expo	N+SPT	L+AT-RPT	L+AT	N+SPT	L+SPT	L+MST	L+CR
				L+SPT	L+AT	L+AT-RPT	L+SPT	N+SPT	L+EDD	L+EDD
				L+RR	N+AT	N+AT	N+RR	L+MDD	N+MST	L+MST
			Normal	N+MDD	N+AT-RPT	L+AT-RPT	N+SPT	L+MDD	L+MST	N+EDD
				L+MDD	L+AT-RPT	N+AT-RPT	L+SPT	N+MDD	N+MST	L+EDD
				N+EDD	N+MST	N+AT	N+PT+WINQ	N+S/RMOP	L+EDD	N+MST
5	95%	6	Uniform	N+SPT	L+AT-RPT	L+AT-RPT	L+SPT	L+MDD	N+MST	L+CR
				L+SPT	N+AT-RPT	N+AT-RPT	N+SPT	N+MDD	L+MST	L+S/RMOP
				L+MDD	L+AT	L+AT	N+RR	L+S/RMOP	L+EDD	L+EDD
			Expo	N+SPT	L+AT-RPT	L+AT	N+SPT	L+SPT	N+MST	L+EDD
				L+SPT	L+AT	L+AT-RPT	L+SPT	N+SPT	L+MST	N+EDD
				N+RR	N+AT	N+AT	Random+SPT	L+MDD	L+EDD	L+MST
			Normal	N+MDD	N+AT-RPT	L+AT-RPT	L+SPT	L+MDD	N+MST	N+EDD
				N+EDD	L+AT-RPT	N+AT-RPT	N+SPT	N+MDD	L+MST	L+CR
				L+MDD	L+AT	N+AT	N+MDD	N+S/RMOP	N+EDD	N+MST

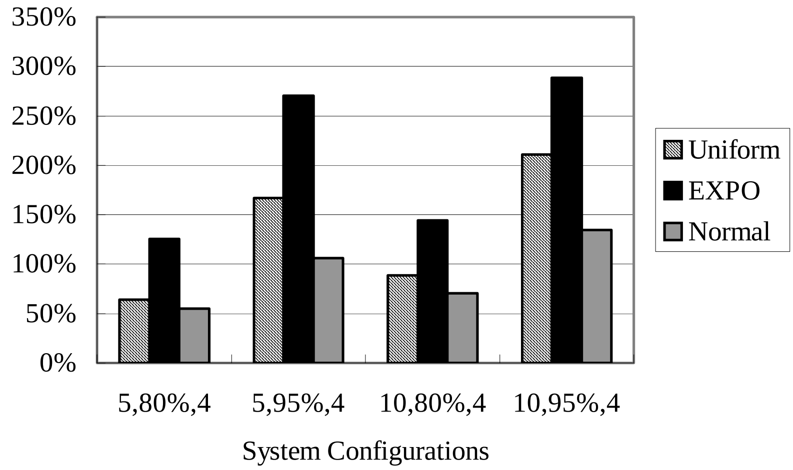

Figure 1. Percentage performance difference in $\overline{F}$ between the best and worst dispatching rule-pairs (including *Random* machine-selection rule)

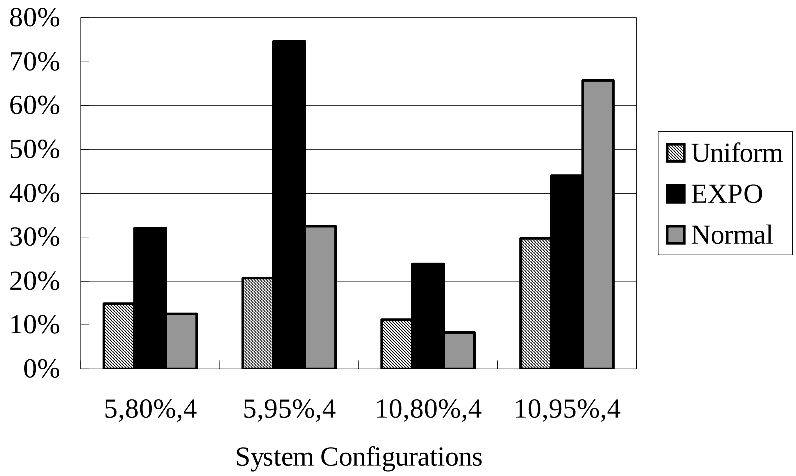

Figure 2. Percentage performance difference in $\overline{F}$ between the best and worst dispatching rule-pairs (excluding *Random* machine-selection rule)

4.3 Comparisons among Different Production Systems

In addition to the analysis of rule-pairs in an open shop, this study also compares the best job-dispatching rule of an open shop to that of a job shop or a flow shop with similar system configurations. Table 7 outlines the comparison results. The results show that under the same performance criterion, the best job-dispatching rules may not be the same for these three different types of production system. Especially, MST, a job-dispatching rule that never is the best in a flow or job shop no matter what performance criterion is chosen, is among the best in an open shop when tardiness related criteria are considered. On the other hand, PT+WINQ+SL, a job-dispatching rule that performs well in a flow or job shop under maximum tardiness and tardiness variance criteria, is not listed in the top three best job-dispatching rules for an open shop.

Table 7. The best job-dispatching rules for different types of manufacturing systems

	$\bar{F}$	F_{max}	σ_F^2	%T	$\bar{T}$	T_{max}	σ_T^2
Flow Shop	SPT MDD PT+WINQ	SPT FIFO MDD AT AT-RPT PT+WINQ+AT	PT/TIS (PT+WINQ)/TIS	SPT RR	MDD RR COVERT CR OPCR	RR PT+WINQ+SL	RR PT+WINQ+SL
Job Shop	SPT RR PT+WINQ	AT-RPT PT+WINQ+AT PT+WINQ+AT+SL	AT-RPT PT+WINQ+AT PT+WINQ+AT+SL	MOD SPT RR FIFO	MOD RR FIFO	RR PT+WINQ+SL	RR PT+WINQ+SL PT+WINQ+AT+SL
Open Shop	SPT MDD EDD RR PT+WINQ+AT	AT-RPT AT MST FIFO	AT-RPT PT+WINQ+AT AT	MDD EDD MST SPT RR	MDD EDD MST SPT S/RMOP	MST EDD MDD AT-RPT S/RMOP	MDD EDD MST CR S/RMOP

5. Conclusions

Dispatching problems in an open shop is more complicated than others. It not only takes into account the job priority at each machine but also the selection of the next machine for any jobs leaving the current machine. The control mechanism for scheduling in an open shop can be characterized by a dispatching rule-pair that consists of a machine-selection rule and a job-dispatching rule. In this study, simulation models for open shops are constructed and run to collect dada for comparing the performance of different dispatching rule-pairs. There are a total of 39 dispatching rule-pairs considered in this study. Through carefully designed experiments, this study finds that the choice of the best dispatching rule is affected by the utilization factor, due-date factors, and the process time distribution at each station. This study also finds that machine-selection rule is an important factor in the system's overall performance. Using NINQ or LINQ is better than using *Random* as the machine-selection rule for an open shop. In general, using NINQ as the machine-selection rule can minimize the mean flowtime of jobs while using LINQ can minimize mean tardiness of jobs for most cases.

All rule-pairs are compared to each other under different performance criteria. For mean flowtime criteria, NINQ+SPT and LINQ+SPT are the best choices in general while NINQ+MDD and NINQ+EDD are also good for systems with uniformly distributed process times. Meanwhile, NINQ+AT-RPT and LINQ+AT-RPT both outperform others in maximum flowtime and flowtime variance criteria. On the other hand, MDD, EDD, and MST job-dispatching rules coupled with NINQ or LINQ machine-selection rules create dispatching rule-pairs that can minimize tardiness related criteria of jobs. The mean flowtime of jobs for systems using the worst dispatching rule-pair is 55% to 289% more than that of using the best dispatching rule-pair. The differences reduce to 8% to 75% if excluding *Random* machine-selection rule. In other criteria, the performance difference between systems using the best and the worst dispatching rule-pair ranges from 4 to well over 70 times. Finally, the best dispatching rule for an open shop is different from that of a job shop or a flow shop with similar system configurations.

Although this study explored existing dispatching rule-pairs in open shops using simulation analysis, there remain many research topics to be studied. For example, new dispatching rule-pairs are needed to further utilize open shops' capability. In addition, the transient behavior of an open shop when any dynamic events occurred need to be studied to see if any dispatching rule-pair performs better than others under the circumstance. Finally, as in DRC (Dual-resource Constrained) job shops, worker-dispatching rules need to be studied when labor shortage occurred in an open shop.

Acknowledgments

This research is supported by the Research Grant number NSC 91-2213-E-324-019 from the National Science Council, Taiwan.

References

[1] Hunsucker, J. L. and Shah, J. R. 1994. Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment, *European Journal of Operational Research,* 72: 102-114.

[2] Sarper, H. and Henry, M. 1996. Combinational evaluation of six dispatching rules in a dynamic two-machine flow shop, *Omega*, 24, 1: 73-81.

[3] Barrett, R. and Kadipasaoglu, S. N. 1990. Evaluation of dispatching rules in a dynamic flow shop, *Production and Inventory Management Journal*, 31, 1: 54-58.

[4] Rajendran, C. and Holthaus, O. 1999. A comparative study of dispatching rules in dynamic flow shops and job shops, *European Journal of Operational Research,* 116: 156-170.

[5] Ramasesh, R. 1990. Dynamic job shop scheduling: a survey of simulation research, *Omega,* 18, 1: 43-57.

[6] Waikar, A. M., Sarker, B. R. and Lal, A. M. 1995. A comparative study of some priority dispatching rules under different job shop loads, *Production Planning and Control*, 6, 4: 301-310.

[7] Holthaus, O. 1997. Design of efficient job shop scheduling rules, *Computers and Industrial Engineering,* 33, 1/2: 249-252.

[8] Raghu, T. S. and Rajendran, C. 1993. An efficient dynamic dispatching rule for scheduling in a job shop, *International Journal of Production Economics,* 32: 301-313.

[9] Holthaus, O. and Rajendran, C. 1997. Efficient dispatching rules for scheduling in a job shop, *International Journal of Production Economics,* 48: 87-105.

[10] Holthaus, O. and Rajendran, C. 2000. Efficient jobshop dispatching rules: further developments, *Production Planning and Control*, 11, 2: 171-178.

[11] Liu, K. C. 1998. Dispatching rules for stochastic finite capacity scheduling, *Computers and Industrial Engineering,* 35, 1: 113-116.

[12] Gonzalez, T. and Sahni, S. 1976. Open shop scheduling to minimizing finish time,

Journal of the ACM, 23: 665-679.

[13] Barker, K. R. and Kanet, J. J. 1983. Job shop scheduling with modified due dates, *Journal of Operations Management*, 4, 1: 11-22.

[14] Jayamohan, M. S. and Rajendran, C. 2000. New dispatching rules for shop scheduling: a step forward, *International Journal of Production Research*, 38, 3: 563-586.

[15] Kher, H. V. and Fry, T. D. 2001. Labour flexibility and assignment policies in a job shop having incommensurable objectives, *International Journal of Production Research*, 39, 11: 2295-2311.

[16] Law, A. M. and Kelton, W. D. 1991. *Simulation Modeling and Analysis*, McGraw-Hill, New York.

[17] Kelton, W. D., Sadowski, R. P., and Sadowski, D. A. 2002. "*Simulation with Arena" (2nd Edition)*, McGraw-Hill, New York.

[18] Sue, T. W. and Liaw. C. H. 1999. Minimizing the total weighted flowtime for the open shop scheduling problem, *Proceedings of the Chinese Institute of Industrial Engineers-National Conference (CD-ROM)*.