

# **SCHEDULING RULES FOR A SMALL DYNAMIC JOB-SHOP: A SIMULATION APPROACH**

Dileepan, P. & Ahmadi, M.

University of Tennessee at Chattanooga, Faculty of Operations Management,  
615 McCallie Avenue, Dept. 6156, Chattanooga, TN 37403, USA  
E-Mail: mohammad-ahmadi@utc.edu

## **Abstract**

The objective of this paper is to study a selected set of scheduling criteria commonly used in scheduling research and develop a set of simple rules that can be easily implemented in relatively small dynamic job-shops. Simulation models were developed for two dynamic production shop environments, namely, a job-shop with jumbled flow and a job-shop with limited jumbled flow. Five scheduling criteria were used with each of the three models. They are, (i) Slack per Remaining Operations (SPRO), (ii) Priority Ratio (PR), (iii) Earliest Due Date (EDD), (iv) Total Work Content Remaining (TWKR), and (v) Smallest Processing Time (SPT). In each of these cases, performance statistics relating to work-in-process (WIP) inventory and job lateness were gathered. The results show that in most cases, simple rules such as SPT and EDD outperform the other rules. SPT rule minimizes the performance metrics that measure WIP inventory levels and the EDD rule minimizes the performance metrics that measure deviation of job completion time from the promised due date.

(Received in February 2010, accepted in July 2010. This paper was with the authors 2 months for 1 revision.)

**Key Words:** Scheduling, Job-Shop, Simulation

## **1. INTRODUCTION**

In the last few decades the problem of scheduling jobs in a manufacturing environment has received a lot of attention from researchers. A wide variety of manufacturing environments have been studied, differing in several characteristics such as the number of machines in the shop, flow of jobs through the shop, the job arrival patterns, and scheduling criteria. Much of the early efforts were restricted to single-machine production shops with a static set of jobs. The performance measures studied were a function of job completion time and job lateness. As the interest in finding and implementing the best practices in manufacturing grew, the types of scheduling problems studied also grew, both in breadth and depth. Consequently, the literature today is filled with articles on a wide array of scheduling problems, each offering a scheduling algorithm or heuristic for a given scheduling environment and a set of performance criteria. However, there is little evidence to show that any of this theoretical knowledge is applied in practice. The reasons for this may include the overabundance of scheduling literature dealing with every obscure scheduling problem that can be conceived and the absence of practitioner-ready synthesis of the results. In this paper, we try to address this problem by experimenting with a selected set of scheduling criteria commonly used in scheduling research and develop a set of simple rules that a practitioner can apply and benefit.

Among the earliest scheduling literature are Conway et al. [1] and Baker [2]. Much of their work was for a static set of jobs in a single machine job-shop. Scheduling in a dynamic job-shop is a complex endeavour that requires continued balancing of many factors such as machine breakdown, absenteeism, availability of material, availability of tools. In this environment, the challenge of scheduling is to find at each workstation of the job-shop the next job that must be processed so that the work-in-process inventory is kept to a minimum

and the jobs are completed as close to the due dates as possible. Much of the scheduling research is focused on finding schedules, through some analytical means, that are optimal with respect to one or more performance criteria. Further, these approaches are often applicable only for a narrow sliver of job-shops due to the assumptions that are required for the methodologies to work. However, most practitioners will benefit if a small set of scheduling rules that are easy to apply can be shown to be effective, with respect to commonly used performance metrics, under a range of job-shop environments. Mohanasundaram et al. [3] present a simulation study of a dynamic assembly shop. They present scheduling rules for minimizing the maximum standard deviation of flow time and staging delay and the maximum conditional tardiness and standard deviation of conditional tardiness. Thiagarajan and Rajendran [4] present another simulation study of a dynamic assembly shop. They present dispatching rules for minimizing the sum of weighted tardiness and earliness, and the sum of weighted tardiness, earliness, and flow time.

As far back as 1999, Kosturiak and Gregor [5] argued for a new kind of decision support system for rapidly changing environments. They summarized how discrete-event simulation can be used in the design, operation, and continuous improvement of complex manufacturing and logistical systems. They discussed the results of this approach in logistics, flexible manufacturing, electrical engineering industry, furniture assembly and tire manufacturing. Hung et al. [6], in their paper present a new modelling approach for realistic simulation of supply-chains. They present a case study to illustrate the applicability of the simulation model. Chryssolouris et al. [7] used simulation with the help of commonly used assignment rules for scheduling of a simple manufacturing system and studied the results to propose a new method for scheduling. Mönch [8] presented a simulation approach for applying benchmarking efforts for production control approaches to complex manufacturing systems that are characterized by a large number of products, changing product mix, sequence-dependent set-up times, unrelated parallel machines, a mix of different process types, and internal and external disturbances. Gupta and Sivakumar [9] present a simulation study of a single machine with due dates and sequence-dependent setup times, where the multiple objectives are to minimize average cycle time and average tardiness and to maximize machine utilization. Vinod and Sridharan [10] present a simulation-based experimental study of scheduling rules for scheduling a dynamic job shop in which the setup times are sequence dependent. Their results indicate that setup-oriented rules provide better performance than ordinary rules. There clearly is an abundance of attempts to apply simulation analysis for a range of scheduling problems. In this paper, we present a practical simulation approach to develop simple sequencing rules for a general purpose job shop environment.

## **2. THE STUDY**

In this study, we consider a relatively small job-shop with moderate to complex levels of jumbled flow. The objective is to develop simple job sequencing rules that are easy to follow and that will result in efficient operation.

The number of work centers in a job-shop and the degree of routing diversity among the jobs processed are two of the most important factors that determine the complexity of job-shop scheduling. Further, the degree of routing divergence itself is a function of the number of work centers in the job-shop. The simplest of job-shops is one with just one machine and hence just one possible routing. In this case, the Smallest Processing Time (SPT) scheduling rule is known to minimize the sum completion times of a set of static jobs. The sum of completion times of jobs is often used as a surrogate measure for WIP inventory. Similarly, the Earliest Due Date (EDD) scheduling rule is known to minimize the maximum lateness of a static set of jobs.

As the number of work centers of the job-shop is increased, the possibility of routing divergence of jobs will increase. For a given job-shop with  $n$  work centers, the simplest of the routings is one where all the jobs flow through the job-shop in the same sequence. Such a job-shop is called a flow shop. The most complex case is one where a given job could follow any one of the  $n!$  possible routings. Not all of the  $n!$  routings will be feasible in a practical environment. In general, the complexity of the scheduling task will increase with the increasing number of possible routings. In this study, we have chosen two levels of routing divergence. In the first case, relatively few possible routings are considered for each job. Each arriving job is assigned one of the possible routings. After completion of a job at a given machine, each job is routed to the waiting line of the next machine in the job routing assigned to the job. This scenario is referred to as “Moderate”. In the second scenario a relatively large number of possible routings are considered for the jobs. The large number of possible routings results in a job-shop with relatively severe jumbled flow compared to the first scenario. Therefore, this scenario is referred to as “Complex” job-shop.

Job-shops are characterized by factors such as machines, layout, labour, and customer orders. A job-shop is one with general-purpose machinery bought from machine-tool manufacturer catalogues that is laid out on the shop floor organized by functionality. A highly trained and skilled labour force uses these machines to produce products that are customized for individual customer requirements. Since customer orders are characterized by unique features and requirements, the production process routings are often unique as well, resulting in a jumbled flow of customer orders through the shop floor. Examples of job-shops include manufacturing custom products for a range of industries such as construction, performance automobiles and consumer hardware, foundries producing castings to customer design, servicing complex equipment, and custom furniture.

In such an environment, each scheduling decision at each workstation will have a cascading effect all through the shop. No analytical methods exist that can be used to develop optimal scheduling on a dynamic basis. However, some analytical results are available that ignore the interdependencies among various workstations in the job-shop. They include such scheduling rules as Slack Per Remaining Operation (SPRO), Priority Ratio (PR), Earliest Due Date (EDD), Total Work Remaining (TWKR), and Smallest Processing Time (SPT). Description of these rules can be found in any Operations Management textbook.

For the purpose of this analysis, job-shops with  $n = 3$  work centers,  $n = 7$  work centers, and  $n = 10$  work centers are considered. In each of these cases, the two routing scenarios, “Moderate” and “Complex” are studied. These are summarized in Table I.

Table I: Summary of scenarios.

No. of work centers	Moderate	Complex
3	2 routings	4 routings
7	5 routings	10 routings
10	10 routings	20 routings

For each of the six scenarios summarized in Table I, five queuing rules are considered, namely, Slack Per Remaining Operation (SPRO), Priority Ratio (PR), Earliest Due Date (EDD), Total Work Remaining (TWKR), and Smallest Processing Time (SPT). As the jobs arrive at a given work center, they are prioritized by the corresponding rule described in Table II.

Table II: Scheduling rules.

Rule	Description	Priority
SPRO	(Due date – Total work content remaining) / Number of remaining routing steps	By the least value
PR	Due date / Total work content remaining	By the least value
TWKR	Total work remaining	By the least value
EDD	Due date	By the least value
SPT	Processing time at the work center	By the least value

The first three rules, namely, SPRO, PR, and TWKR are multiple-dimension rules as they incorporate information about not just the current work station but all the remaining work stations as well. The last two rules, SPT and EDD are single-dimension rules as they take into account only the information about the current work station.

Under each job-shop routing scenario and scheduling rule, statistics relating to five performance metrics are observed. Two of these five measure the level of WIP in the job-shop. They are average completion time of jobs and average number of jobs in the job-shop. The remaining three measure the divergence between actual job completion time and the promised due date. They are average lateness/earliness, average lateness, and percentage of late jobs.

### **3. SIMULATION MODELLING**

The simulation model is developed using ProModel® software for each of the six job-shop and routing scenarios summarized in Table I. For each model, a dynamic set of jobs are generated with the time between consecutive job arrivals generated from an exponential probability distribution with a mean of 1 hour. The routing for each job is assigned with equal probability. For example, for the case of a complex routing scenario in a job-shop with 7 work centers, each of the ten possible routings has 10 % probability of being assigned to a given job. Next, batch size for each job is generated from a uniform probability distribution between 80 and 120 units. The processing time for each job at each of the work centers in the assigned routing is determined by: setup time + run time  $\times$  batch size. The tables in Appendix show all the setup times and run-times used in this analysis. The setup time and runtimes are arbitrarily chosen. These may be substituted with actual times in a given environment. The due dates are determined using a method similar to the ones used by Fisher [11].

Finally, a due date is calculated for each job from a uniform distribution between  $\psi/2(\tau - R/2)$  and  $\psi/2(\tau + R/2)$ , where  $\psi$  is a maximum load among the work centers in the job routing at the time of arrival of the job + processing time of the job in work centers that follow the one with the maximum load. The value of  $\psi$  is an indicator of the backlog along the routing of the given job. The due dates determined as a function of  $\psi$  are likely to be realistic. A value between 0 and 1 is used for the tardiness factor  $\tau$ . The smaller the value of  $\tau$ , the smaller the value of due dates will be. A value between 0 and 1 is chosen for  $R$  as well. The larger the value of  $R$ , the wider the range of due dates among the generated jobs. Five values for  $R$  are chosen for this experimentation, namely, 0.2, 0.4, 0.6, 0.8, and 1.0. For each value of  $R$ , the value of  $\tau$  is chosen from the list of 0.2, 0.4, 0.6, 0.8, 1.0, such that  $\tau > R/2$ , so that negative due dates are not generated. For example, for  $R = 0.6$ ,  $\tau$  is chosen from 0.4, 0.6, 0.8, and 1.0. The value  $\tau = 0.2$  is not considered as it is smaller than  $0.6/2 = 0.3$ . Thus, 21 combinations of  $t$  and  $R$  values, as shown in the Table III, are considered in this analysis.

Table III: Feasible combination of  $R$  and  $\tau$  values.

$R$	$\tau$ values
0.2	0.2, 0.4, 0.6, 0.8, 1.0
0.4	0.2, 0.4, 0.6, 0.8, 1.0
0.6	0.4, 0.6, 0.8, 1.0
0.8	0.4, 0.6, 0.8, 1.0
1.0	0.6, 0.8, 1.0

For each of the six routing scenario and Job-shop size combinations (Table I), and the five scheduling rules (Table II), the 21 sets of  $R$  and  $\tau$  values (Table III) are used to simulate job arrivals and job due dates. Thus, in this study we consider  $6 \times 21 = 126$  unique job-shop environments with conditions ranging from moderate to complex job routings, lenient to tight and narrow to broad range of due date values.

Each of the 126 scenarios is simulated with the five scheduling criteria summarized in Table II. The simulation is run for a warm-up period of 1000 hours during which no performance statistics are collected. The simulation is then continued for another 5000 hours during which a range of performance related statistics are gathered. They are (i) Average completion time, (ii) Average number of jobs in the job-shop, (iii) Average lateness/earliness, (iv) Average lateness, and (v) Percentage of late jobs. The first two of these five metrics are surrogate measures for the level of WIP inventory in the job-shop. The last three measure the delivery performance with respect to the promised due date. Each simulation is repeated 100 times, and the averages of the performance measures are collected from the 100 replications.

#### **4. ANALYSIS OF SIMULATION RESULTS**

For each of the 126 scenarios, 100 replications of the simulation were run for each of the five scheduling criteria. For each of the six scenarios presented in Table I, the average of the performance metrics from the 21 problem set combinations of Table III was used to compare the five scheduling rules. Table IV presents the scheduling rules that resulted in the minimum values for the two WIP related measures, namely, average completion time and the average number of jobs in the shop.

Table IV: Simulation results – best rules for minimizing WIP inventory.

Number of machines	Number of routes	Average completion time	Average number of jobs in the shop
$n = 10$	Moderate	SPT	SPT
	Complex	SPT	SPT
$n = 7$	Moderate	SPT	SPT
	Complex	SPT	SPT
$n = 3$	Moderate	SPT	SPT
	Complex	SPT	TWKR

The simple rule of SPT outperformed all other rules across the board with respect to the average completion time. Further, SPT outperformed all other rules with respect to the average number of jobs in the shop except for the case of “Complex” routing and  $n = 3$

machines for which the TWKR rule is the best. These results show that the single-dimension SPT rule can be an effective scheduling rule for small operations for minimizing overall work-in-process inventory. TWKR rule may also be used in really small operations.

Table V presents the results for performance metrics related to the deviation between job completion time and the promised due date. Unfortunately, no single scheduling rule stands out in Table V. However, this analysis shows that the PR rule is not the best rule under any of the scenarios considered. In fact, in most cases, the PR rule results in the worst values of the three measures given in Table V, among all the five rules considered in this study.

Table V: Simulation results – best rules for minimizing lateness/earliness.

Number of machines	Number of routes	Average lateness	Percentage of late jobs	Average absolute lateness/earliness
$n = 10$	Moderate	SPT	EDD	SPT
	Complex	TWKR	EDD	SPRO
$n = 7$	Moderate	SPT	TWKR/SPT	EDD
	Complex	SPT	SPT	EDD
$n = 3$	Moderate	EDD	SPT	EDD
	Complex	EDD	EDD	EDD

This is illustrated by the two graphs of the results shown in Figs. 1 and 3. Fig. 1 shows the average lateness metric for the case of  $n = 10$  work centers and “Complex” routing. Fig. 3 shows the percentage of late jobs for the case of  $n = 7$  and “Moderate” routing. Both graphs show the PR rule results in poorest values of these measures compared to the other four rules.

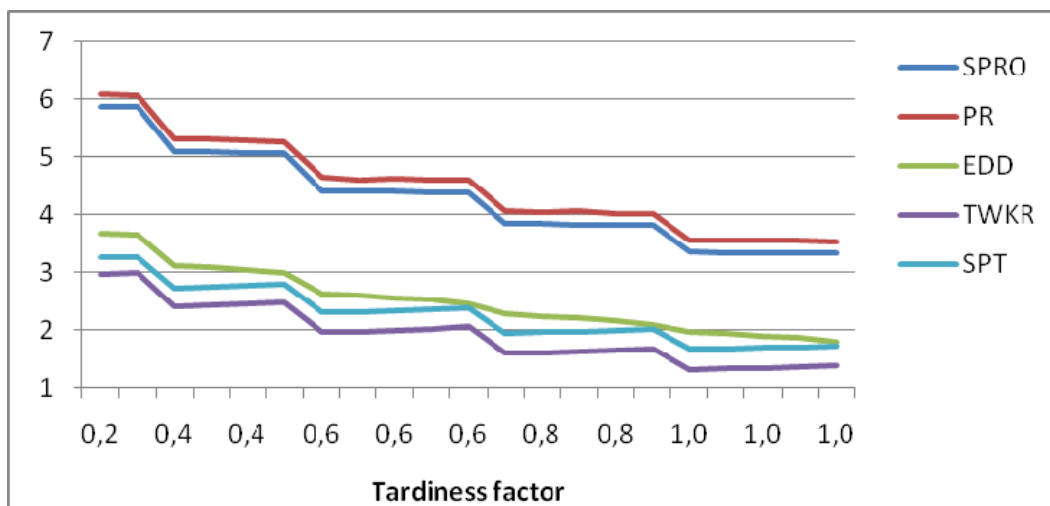


Figure 1: Average lateness for  $n = 10$  / Complex routing.

Further, Table V shows that the two single-dimension rules (SPT and EDD) provide the best results for all the scenarios except two cases. The first one is for the average lateness for the case of  $n = 10$  and complex routing. The results for this case are shown in Fig. 1. The multi-dimensional TWKR rule provides the best results. However, the single-dimensional SPT rule is a close second best for this scenario. Also, for the average absolute lateness metric for the case of  $n = 10$  and “Complex” routing, the multi-dimensional SPRO rule provides the best results. The entire set of the results is shown in the graph given in Fig. 2. In this case, we

find that no single rule is uniformly the best across all value of the tardiness factor  $\tau$ . The figure shows that the single-dimension SPT rule is the best for problems with  $t \leq 0.6$ . Therefore, it is perhaps not unreasonable to conclude that in most of the scenarios considered, a single-dimensional rule is the best for problems with  $t \leq 0.6$ . Therefore, it is perhaps not unreasonable to conclude that in most of the scenarios considered, a single-dimensional rule is just as good or better than any of the multi-dimension rules with respect to performance metric related to divergence between job completion time and promised due date.

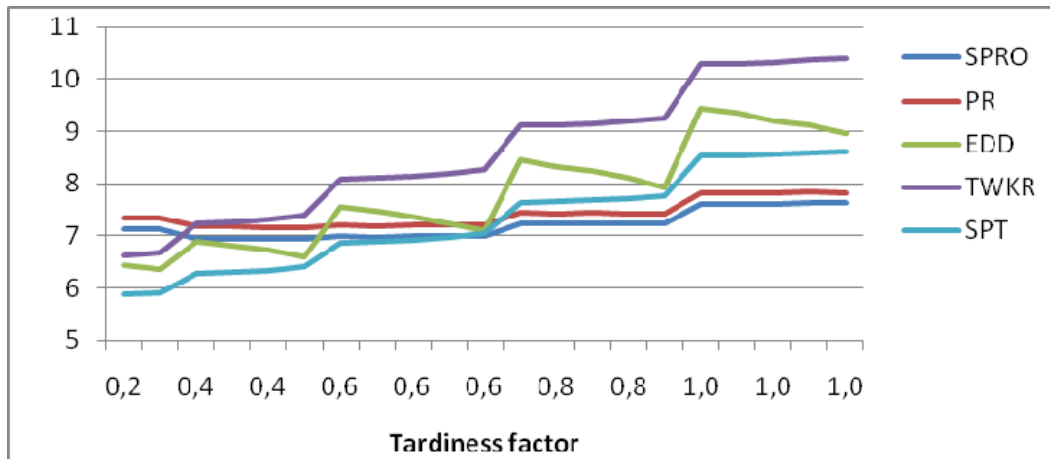


Figure 2: Average absolute lateness/earliness for  $n = 10$  / Complex routing.

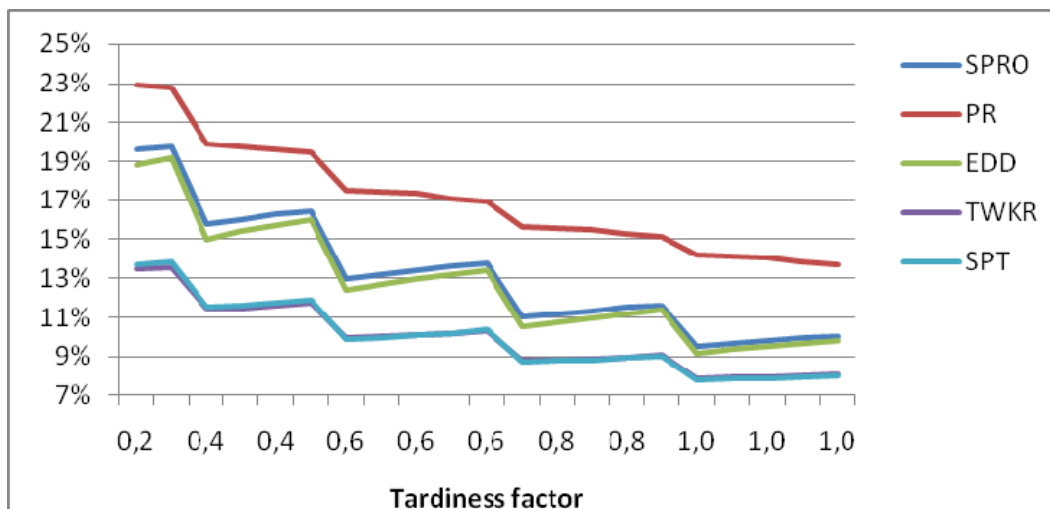


Figure 3: Percentage of late jobs for  $n = 7$  / Moderate routing.

## 5. CONCLUSION

Scheduling in a job-shop is a complex task requiring the balancing of a wide range of resources. The two important scheduling criteria are minimization of work-in-process inventory and the minimization of lateness/earliness. There is an abundance of scheduling research. However, these results are applicable only within the narrow range of assumed operating conditions. Further, the more complex a scheduling rule is, the less likely it is to be adopted by practitioners. This is especially true for relatively small job-shops as they typically would lack the financial and technical resources to fully take advantage of the benefits of complex scheduling rules. This study is an attempt to develop simple rules that can be easily

implemented. Relatively small job-shops with up to 10 work centers were simulated under a range of conditions. An analysis of the results show that in most cases simple single-dimension rules such as SPT and EDD would provide reasonable to best outcomes with respect to performance metrics. The SPT rule invariably works best for minimizing the overall reduction of WIP inventory across the job-shop. For the case of job lateness/earliness, the SPT and EDD rules provide reasonable results and are just as good as other multi-dimensional rules.

## **REFERENCES**

- [1] Conway, R. W.; Maxwell, W. L.; Miller, L. W. (1967). *Theory of Scheduling*, Addison-Wesley, Reading
- [2] Baker, K. R. (1974). *Introduction to Sequencing and Scheduling*, Wiley, New York
- [3] Mohanasundaram, K. M.; Natarajan, K.; Viswanathkumar, G.; Radhakrishnan, P.; Rajendran, C. (2003). Scheduling rules for dynamic shops that manufacture multi-level jobs, *Computers & Industrial Engineering*, Vol. 44, No. 1, 119-131
- [4] Thiagarajan, S.; Rajendran, C. (2005). Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flow time of jobs, *Computers & Industrial Engineering*, Vol. 49, No. 4, 463-503
- [5] Kosturiak, J.; Gregor, M. (1999). Simulation in production system life cycle, *Computers in Industry*, Vol. 38, No. 2, 159-172
- [6] Hung, W. Y.; Samsatli, N. J.; Shah, N. (2006). Object-oriented dynamic supply-chain modelling incorporated with production scheduling, *European Journal of Operational Research*, Vol. 169, No. 3, 1064-1076
- [7] Chrysolouris, G.; Giannelos, N.; Papakostas, N.; Mourtzis, D. (2004). Chaos Theory in Production Scheduling, *CIRP Annals - Manufacturing Technology*, Vol. 53, No. 1, 381-383
- [8] Mönch, L. (2007). Simulation-based benchmarking of production control schemes for complex manufacturing systems, *Control Engineering Practice*, Vol. 15, No. 11, 1381-1393
- [9] Gupta, A. K.; Sivakumar, A. I. (2005). Single machine scheduling with multiple objectives in semiconductor manufacturing, *The International Journal of Advanced Manufacturing Technology*, Vol. 26, No. 9-10, 950-958
- [10] Vinod, V.; Sridharan, R. (2008). Scheduling a dynamic job shop production system with sequence-dependent setups: An experimental study, *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 3, 435-449
- [11] Fisher, M. L. (1976). A dual algorithm for the one-machine scheduling problem, *Mathematical Programming*, Vol. 11, 229-251

## **APPENDIX: Setup and Run-times used in the simulation analysis**

### **Three machine ( $n = 3$ ) problems**

Number of routes	Routing	Machine					
		1	2	3	1	2	3
Moderate		Setup time (minutes)			Run-time per unit (minutes)		
	1	15	20	25	0.5	0.4	0.3
	2	20	10	25	0.2	0.5	0.3
Complex		Setup time (minutes)			Run-time per unit (minutes)		
	1	15	20	25	0.5	0.4	0.3
	2	20	10	25	0.2	0.5	0.3
	3	15	20	30	0.4	0.3	0.2
	4	20	10	30	0.3	0.6	0.3

**Seven machine ( $n = 7$ ) problems**

Number of routes	Routing	Machine						
		1	2	3	4	5	6	7
Moderate		<b>Setup time (minutes)</b>						
	1	15	20			25	30	10
	2	35		25		15		15
	3	25	10	30	30		15	15
	4	15		20	30		20	10
	5	20	25	20		20		15
		<b>Run-time per unit (minutes)</b>						
	1	0.2	0.2			0.4	0.3	0.4
	2	0.5		0.4		0.2		0.3
	3	0.4	0.3	0.2	0.4		0.2	0.6
	4	0.1		0.3	0.5		0.2	0.5
5	0.3	0.2	0.2		0.2		0.4	
Complex		<b>Setup time (minutes)</b>						
	1	15	10		15	15	10	25
	2	15	30	30	25	25	20	
	3	20	25	20	10	10	25	
	4	30	10	25		10	10	10
	5	10	25	30	15	20		30
	6	10		25	25	25	15	30
	7	20		25	10	15	10	25
	8	20	30	30	20		20	10
	9	15	25	10	10	20	20	
	10	25	15	25	25	10		10
		<b>Run-time per unit (minutes)</b>						
	1	0.6	0.3		0.5	0.5	0.3	0.5
	2	0.4	0.2	0.5	0.6	0.2	0.3	
	3	0.3	0.3	0.2	0.2	0.3	0.6	
	4	0.5	0.4	0.2		0.6	0.5	0.4
	5	0.5	0.4	0.4	0.2	0.2		0.5
	6	0.5		0.4	0.5	0.2	0.3	0.3
	7	0.2		0.5	0.5	0.6	0.3	0.5
	8	0.6	0.5	0.3	0.3		0.4	0.5
9	0.5	0.3	0.3	0.5	0.4	0.2		
10	0.2	0.5	0.2	0.3	0.4		0.6	

**Ten machine ( $n = 10$ ) problems**

Number of routes	Routing	Machine										
		1	2	3	4	5	6	7	8	9	10	
Moderate		<b>Setup time (minutes)</b>										
	1	15		25	20	20	25	30		15	20	
	2	15	20	25	20		15		30	30	15	
	3	30	30	20	20	15	15		25	15		
	4	20		25		15	30	20	15	15	20	
	5	30		25	15	30	30		20	25	25	
	6	25	30	20	15	20	15	30	20	30	20	
	7	15	30	15	20	25		15	30	30		
	8	15	25	20	30	20	30			30	20	
	9	15	20	20	30	15			15	25	20	
	10	30		15	15		15	30	30	30	20	
			<b>Run-time per unit (minutes)</b>									
	1	0.4		0.3	0.4	0.2	0.3	0.3		0.2	0.5	
	2	0.2	0.5	0.3	0.2		0.3		0.2	0.3	0.3	
	3	0.2	0.3	0.4	0.5	0.2	0.5		0.3	0.3		
	4			0.4	0.3	0.2	0.2	0.4	0.3	0.2	0.2	
	5	0.2		0.4	0.3	0.5	0.3		0.2	0.4	0.5	
	6		0.5	0.4		0.3	0.5	0.2	0.5	0.3	0.5	
	7	0.5	0.2	0.2	0.2	0.5		0.2	0.2	0.4		
	8	0.4	0.5	0.4	0.5	0.5	0.4			0.5	0.2	
9	0.3	0.3	0.2	0.3	0.4			0.5	0.2	0.5		
10	0.4		0.3		0.4	0.4	0.5	0.5	0.3	0.3		
Complex		<b>Setup time (minutes)</b>										
	1	15	20	20	30	15			15	25	20	
	2	15	20	25	15		25	15	15	25		
	3	30	25	30	30	25	25	20	15			
	4	20	30	30	15	25	20			15	15	
	5	30	20		30	30	25		15	15	15	
	6	15		25	25	25	15	25	25		15	
	7	15	20	25	20		15		30	30	15	
	8	15		25	20	20	25	30		15	20	
	9	15	30	15	20	25		15	30	30		
	10	30	30	20	20	15	15		25	15		
	11	30		25	15	30	30		20	25	25	
	12	15	25	20	30	20	30			30	20	
	13		30	20	30	25	20	30	25		15	
	14		25	25	20	20	25		25	30	30	
	15			30	30	30	20	25	15	30	25	
	16	20		25		15	30	20	15	15	20	
	17	30		15	15		15	30	30	30	20	
	18		30	25		25	30	15	20	30	25	
	19	25	30	20	15	20	15	30	20	30	20	
20	15	30			25	30	20	25	20	30		

(continued)											
Complex	Run-time per unit (minutes)										
	1	0.3	0.3	0.2	0.3	0.4			0.5	0.2	0.5
	2	0.3	0.4	0.5	0.5		0.2	0.3	0.3	0.4	
	3	0.4	0.2	0.3	0.2	0.5	0.4	0.5	0.5		
	4	0.4	0.5	0.2	0.5	0.4	0.2			0.3	0.2
	5	0.2	0.4		0.2	0.3	0.3		0.3	0.2	0.5
	6	0.2		0.3	0.4	0.5	0.2	0.5	0.5		0.5
	7	0.2	0.5	0.3	0.2		0.3		0.2	0.3	0.3
	8	0.4		0.3	0.4	0.2	0.3	0.3		0.2	0.5
	9	0.5	0.2	0.2	0.2	0.5		0.2	0.2	0.4	
	10	0.2	0.3	0.4	0.5	0.2	0.5		0.3	0.3	
	11	0.2		0.4	0.3	0.5	0.3		0.2	0.4	0.5
	12	0.4	0.5	0.4	0.5	0.5	0.4			0.5	0.2
	13		0.3	0.2	0.5	0.3	0.4	0.2	0.5		0.5
	14		0.3	0.5	0.3	0.3	0.5		0.2	0.3	0.2
	15	0.2	0.4	0.3	0.5		0.5	0.2	0.5	0.2	
	16			0.4	0.3	0.2	0.2	0.4	0.3	0.2	0.2
	17	0.4		0.3		0.4	0.4	0.5	0.5	0.3	0.3
	18	0.5		0.3	0.3		0.2	0.3	0.2	0.5	0.5
	19		0.5	0.4		0.3	0.5	0.2	0.5	0.3	0.5
20	0.4	0.2			0.3	0.2	0.4	0.5	0.2	0.5	