Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

USING SIMULATION BASED APPROACH TO IMPROVE ON THE MEAN CYCLE TIME PERFORMANCE OF DISPATCHING RULES

Chin Soon Chong
Malcolm Yoke Hean Low

D-SIMLAB Programme
Singapore Institute of Manufacturing Technology
71 Nanyang Drive
Singapore 638075, SINGAPORE

Appa Iyer Sivakumar

School of Mechanical and Production Engineering
Nanyang Technological University
Nanyang Avenue
639798, SINGAPORE

Kheng Leng Gay

School of Electrical and Electronics Engineering
Nanyang Technological University
Nanyang Avenue
639798, SINGAPORE

ABSTRACT

This paper presents a simulation based approach to improve on the mean cycle time performance of dispatching rules. The method applies recursive simulation technique on dispatching rules to search for new improved solutions for a set of job shop problems. Due to the nature of the recursive heuristic, performance criteria other than mean cycle time and various dispatching rules can be implemented into the recursive simulation framework without requiring too much effort. The performance of the proposed approach is compared to the underlying dispatching rules as well as a published tabu search procedure. The preliminary results show that the approach outperforms the underlying dispatching rules and is comparable to the tabu search procedure.

1 INTRODUCTION

Intense competition in global market trends of lower product costs, shorter product life cycles and more product variety have resulted in more challenging manufacturing environment. The conflicting objectives of maintaining low inventory level to reduce costs, and quick response to customer demand to remain competitive calls for effective and efficient scheduling. In this respect, there have been extensive studies of scheduling algorithms and heuristics in both static and dynamic job shops for decades by researchers and practitioners (Gere 1966, Blackstone et. al. 1982, Rajendran and Holthaus 1999, Jain and Meeran 1999)

A scheduling problem can be characterized by a set of jobs, each with one or more operations. The operations of a job are to be performed in a specified sequence on specific machines. The objective of scheduling is to determine the job schedules that minimize (or maximize) a measure (or multiple measures) of performance (Rajendran and Holthaus 1999). Due to factorial explosion of possible solutions, job shop scheduling problems are considered to be a member of a large class of intractable numerical problems known as NP-hard (Jain and Meeran 1999). The performance measures that are commonly employed include cycle time (or flow time), machine utilization, throughput rates and inventory level. Of all these, cycle time is considered as a key performance criterion since reducing the cycle time can improve market responsiveness and reduce inventory level of manufacturing system such as semiconductor manufacturing (Sivakumar 1999).

Scheduling techniques range from simple priority dispatching rules such as FIFO (first in first out) and SPT (shortest processing time) to more elaborate techniques such as branch and bound (Brucker et. al. 1994), tabu search (Nowicki and Smutnicki 1996), and shifting bottleneck algorithms (Balas and Vazacopoulos 1998). A dispatching rule is used to select the next job to be processed from a set of jobs waiting in queue of specified machines. It has generally been observed that no single rule performs well for all important criteria of manufacturing system, and the relative performance of the dispatching rules can be influenced by routing of jobs and shop floor configurations (Blackstone et. al. 1982, Rajendran and Holthaus 1999).

Dispatching rules are commonly employed in semiconductor manufacturing such as wafer fabrication due to the complexity of the process (Lee et. al. 2001).

Branch and bound is basically an enumerative strategy that requires excessive computing cost, and is therefore not practical for large scale job-shop scheduling problems. Tabu search and shifting bottleneck algorithms have been reported to be effective in job-shop problems with makespan as the performance criterion (Jain and Meeran 1999). However, these techniques have limited success in the industry due to the following reasons: 1) considerable effort is usually required to incorporate performance criteria other than makespan; 2) many choices have to be carefully made relating to parameter values in order to achieve good solutions. Further, the determination of parameter values requires much effort and is usually done by trial and error; 3) these methods are problem dependent and require multiple runs of the algorithm to obtain good results; 4) results for some of these methods are dependent on the initial solutions (Jain and Meeran 1999).

This work aims to explore a simulation-based scheduling approach to improve on the cycle time performance of dispatching rules through recursive simulation technique. This technique differs from conventional scheduling heuristics as it only relies on the use of general simulation mechanism and priority dispatching rules to perform job shop scheduling. This characteristic enables the technique to be extended to consider a variety of performance criteria with minimum effort, to incorporate different dispatching rules, to minimize work needed in selecting choices, and to eliminate dependency on the initial solutions. The work is at its early stage of research and thus only a simple single-level recursive simulation is investigated.

This paper first reviews the work done in simulation and recursive simulation approaches. Subsequently in Section 3 of this paper, the concept of recursive simulation technique as applied to job shop scheduling problems is addressed. This is then followed by a comparative study on the performance of the simulation-based approach on benchmark problems. The paper finally ends with conclusions and future works in Section 5.

2 SIMULATION-BASED APPROACHES

In a simulation, a system can be imitated by using a computer to study and evaluate a model of the system numerically so as to estimate the characteristics of the system. Simulation has become a widely used tool in operations research and management science (Law and Kelton 1991). Apart from being used as a tool to evaluate system, simulation is also becoming a popular technique for developing production schedules and dispatch lists in a manufacturing environment (Morito and Lee 1997). Simulation based scheduling normally covers short term planning and control, using highly detailed models that are integrated to other information systems. The advantage of using simulation to create a schedule is in the shorter computation times involved as compared to some other techniques (Mazziotti and Horne 1997). Another advantage of using simulation to generate schedules is that simulation can mimic the behavior of the actual system in an intuitive manner which enables users to understand the logic of manufacturing systems (Hopp and Spearman 1996).

In simulation-based scheduling system, two-stage approach is normally employed (Yang and Chang 1998). In the first stage, offline experiments are carried out with various dispatching rules to identify set of rules that can achieve the desired performance based on the condition of the manufacturing system. In the second stage, this set of dispatching rules is used in actual operations to generate schedules of all operations in a system for a fixed time period in future. Complex dispatching rules that integrate management policies, hardware constraints, etc. can be incorporated in simulation based scheduling approach as described in a near-real-time scheduling system (Sivakumar 1999).

The performance of simulation based scheduling approach is dependent on the underlying dispatching rules. In some cases, artificial intelligent techniques such as neural network can be used to select the most appropriate dispatching rules based on the shop floor condition to achieve the desired system objective (Min and Yih 2003). In multi-pass scheduling approach, a set of candidate rules can be evaluated online through a series of simulation to select the best dispatching rule (Jeong and Kim 1998, Yoo et. al. 2004).

A popular operating construct in computer science is recursion, which is a common technique to split a complex problem into its single simplest case. This concept can similarly be applied in simulation to allow for a simulation within a simulation, which is termed as recursive simulation (Gilmer and Sullivan 2000). In this simulation technique, simulated decision makers themselves use simulation to make decision. The recursive simulation runs within a simulation are used to evaluate the possible outcomes given that a decision was made on one way or the other. The number of levels of recursion, the types (deterministic or stochastic) and numbers of runs can be varied.

Very little is reported in literature relating to recursive simulation. The only known publication to date is the use of recursive simulation in military decision-making (Gilmer and Sullivan. 2000, Gilmer and Sullivan 2003, Gilmer 2004). The use of recursive simulation has been demonstrated to improve on the quality of decisions. However, the use of recursive simulation runs can incur high computation costs, especially in the cases of multiple levels of recursion and multiple replications. Nonetheless, with the technological trend of rapid multiplication of computation power and advances in distributed simulation cloning which allows multiple possibilities at a decision point to be tested concurrently (Chen et. al. 2004), the execution speed of recursive simulation can eventually be improved.

In our study of recursive simulation for job-shop scheduling, we have limited the scope to a single level recursive and deterministic simulation run. Multiple levels of recursive simulation runs tend to improve quality of schedules, but the amount of improvement is usually not justifiable for the excessive computation costs incurred. Scheduling problems are usually deterministic and therefore multiple replications of recursive runs are not required.

3 A RECURSIVE SIMULATION-BASED SCHEDULING APPROACH

Figure 1 shows the concept of recursive simulation. The base simulation run is the run invoked by a simulation practitioner. Suppose at a simulation time t_1, a simulated entity needs to make a decision. The entity may use simulation (i.e. first level recursive simulation run) to evaluate various alternatives, and subsequently decides the best alternative to take based on the simulated results. Within the called simulation run, simulated entities can again invoke simulation (second level recursive simulation run) to study different scenarios at decision points.

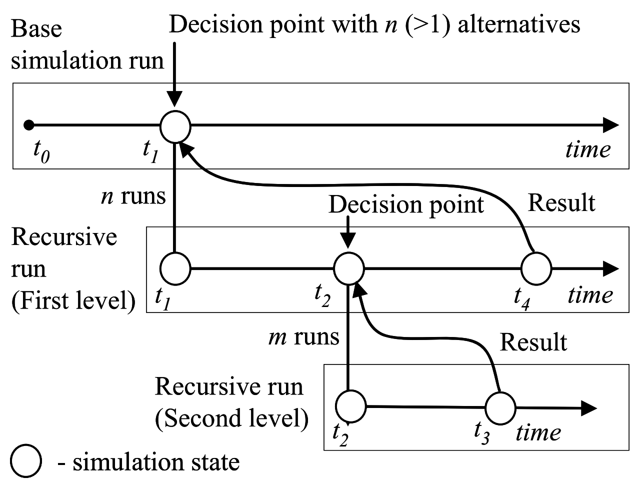

Figure 1: Recursive Simulation

In recursive simulation based scheduling, a decision point may be the time when a machine is ready to process next job in the machine queue. In this case, the machine is the entity and the jobs in the queue are the alternatives. Recursive simulation run is used to evaluate each choice (or job), and the job that gives the best measure of performance, mean cycle time for instance, will be selected at the decision point. For deterministic simulation, the number of recursive simulation runs needed at a decision point is equal to the number of jobs (>1) in the queue. If only a single job is in the queue, the job will be picked for processing without requiring any recursive simulation run. The run length of recursive simulation can be either fixed or based on certain terminating criteria such as a fixed number of jobs complete all its operations.

To better illustrate the our scheduling approach which is limited to single level recursive simulation run, consider an example of heuristics being used to dispatch jobs to machines in the base simulation run and the performance criteria is mean cycle time. Suppose at a decision point in the base simulation run where a machine is ready to load the next job, and there are n jobs in the queue. To determine the best job to load onto the machine, a recursive simulation run (i.e. first level recursive run) is invoked to evaluate the performance (i.e. mean cycle time) for each job in the queue (jobs 1 to n) assuming the simulated scenario that the specific job was already loaded onto the machine in the base simulation. Dispatching rules such as FIFO, SPT, etc. can be used in the recursive simulation to dispatch jobs to machines. The mean cycle time of all jobs in the job shop problems are determined at the end of the recursive simulation run and the result retained for comparison to other recursive runs (i.e. $n - 1$ runs) of other jobs in the queue assuming the job loaded onto the machine in the base simulation run. The job that gives the best mean cycle time among the recursive runs will then be loaded onto the machine at the decision point in the base simulation. This process is repeated for each decision point in the base simulation.

The execution speed of the scheduling approach is apparently dependent on the number of recursive runs and the execution speed of each recursive simulation. The number of recursive runs is proportional to the product of the number of decision points in the base simulation and the average number of jobs in the queue at each decision point. Both the number of decision points and the total number of operations is proportional to the number of jobs in the job shop. Therefore the execution speed is directly proportional to the square of the total number of jobs in the system. The execution speed of each recursive simulation run is affected directly by the underlying dispatching rule and the run length of the simulation. It is noted that if a static job shop with fixed number of jobs in the system is considered, as simulation time progresses in the base simulation, more jobs will complete its operations and therefore fewer jobs are available for selection in the queue. This will result in fewer recursive simulation runs required as simulation time advances.

4 PERFORMANCE COMPARISONS

4.1 Benchmark Problems

The measurement criteria for our study is the mean cycle time (or flow time) which is defined as average amount of time jobs spend in the system. The performance of the proposed scheduling approach is studied by evaluating them on the following benchmark job shop problems (Ganesan et. al. 2004):

- 3 problems from Fisher & Thompson (1963), referred as ftp06, ftp10 and ftp20
- 40 problems from Lawrence (1984), referred as la01 – la40
- 20 problems due to Storer et al. (1992), referred as swv01 – swv20
- 10 problems used by Applegate and Cook (1991), referred as orb1 – orb10
- problems used by Tamada and Nakano (1992), referred as yn1 – yn4
- problems formulated by Adams et al. (1988), referred as abz5 – abz9.

The sizes of these problems range from 6 to 50 jobs and 5 to 20 machines. Larger sizes of shop problems are not considered in the study as past results have concluded that the size of the shop does not affect the relative performance of the dispatching rules, and valid conclusions could be drawn from experiment with relatively small shops (Nanot 1963, Buffa 1968). Earlier results have also shown that the distribution with respect to shape and range of the arrival rate for incoming jobs is not a significant variable in evaluating the relative effectiveness of dispatching rules (Elvers 1974).

4.2 Dispatching Rules and Heuristics

The following dispatching rules and recursive simulation based heuristics are evaluated and compared using the 82 shop problems:

- FIFO (first in first out): This rule chooses the job that has entered the queue at the earliest for loading. FIFO is an effective rule for minimizing the maximum cycle time (Rajendran and Holthaus 1999). This rule is used as a benchmark in the study.
- SPT (shortest process time): This rule selects the job with operation on the machine having the shortest process time. This rule is the most commonly used for job shop scheduling and is found to be very effective in minimizing mean cycle time (Conway 1965, Rochette and Sadowski 1976, Blackstone et. al. 1982).
- SRPT (shortest remaining process time): This rule chooses the job with the least remaining process time, which excludes the current operation under consideration. It is considered in the study since it is found that the rule is effective in minimizing mean cycle time.
- PT+WINQ (process time plus work-in-next-queue): PT is the operation process time of job on the machine, and WINQ is total work content of jobs in the queue of next operation of the jobs. This is a look-ahead rule which is found to be effective in minimizing mean cycle time (Holthaus and Rajendran 1997).
- rFIFO, rSPT, rSRPT and rPT+WINQ: These heuristics are the recursive simulation based implementation of the corresponding dispatching rules. In these heuristics, each job in the queue of a ready machine in the base simulation is enumerated through recursive simulation to determine the best job to load onto the machine. The specified dispatching rules are used in the recursive simulation to dispatch jobs onto machines.
- rDR1: This heuristic evaluates the performance of each job in the queue of specific machine in the base simulation by using a combination of dispatching rules FIFO, SPT, SRPT, PT+WINQ, LPRPT (Least percent remaining process time which is equal to the remaining process time over total process time) and (PT+WINQ)/TIS (where TIS is the time in system of the job) in the recursive simulation runs (number of recursive runs = number of jobs in the queue of the machine x number of dispatching rules) and selects the job that gives the best performance.
- rDR2: This heuristic combines rFIFO, rSPT, rSRPT, rPT+WINQ and rDR1. The heuristic always returns the best schedule among all the recursive simulation based heuristics, but incurs additional computation burden.

In addition, we have modified tabu search implementation described by Nowicki and Smutnicki (1996) to consider mean cycle time instead of makespan for job shop problems. Tabu search is a meta-heuristic based on a local search technique which attempts to exploit the solution space beyond local optimality. Their work is considered as one of the most efficient tabu search implementations for the performance criteria of makespan (Blazewicz et. al. 1996). To cater for mean cycle time criteria instead of makespan, the critical path for each job in the schedule needs to be calculated (Armentano and Scrich 2000). This path determines the completion time of the job, and consequently the cycle time of the job. In comparison with the original work which only determines the critical path of the schedule, the size of neighborhood in the modified work has increased substantially. The size of neighborhood plays an important role in the efficiency of the tabu search procedure. For comparison purpose, we have two versions of the tabu search procedure based on different terminating criteria:

- TB1 (tabu search 1): This procedure uses SPT to obtain an initial solution and searches for new solutions in the neighborhood based on the critical paths of the jobs. The terminating criterion is set to be a specific number of iterations without any improvement such that the run time is in similar range as rDR2 heuristic.

- TB2 (tabu search 2): This procedure is the same as TB1 except for the stopping criterion, which is set to be 5 times the number of iterations of TB1. Hence TB2 will give better performance than TB1 but with much higher computation costs.

The heuristics and procedures were coded using C++ language and run on Windows XP platform with Intel Pentium 400 MHz processor.

4.3 Results

Table 1 summarizes the results on the relative performance of mean cycle time in terms of percentage for the dispatching rules and recursive simulation based heuristics as compared to FIFO. The results show the average, minimum and maximum improvement for the mean cycle time of jobs as well as the number of solutions that is better than the 82 solutions generated by FIFO. For cases of minimum and maximum improvement, the specific job shop problems and their sizes (i.e. number of jobs x number of machines) are also shown in the table.

Dispatching rules SPT, SRPT and PT+WINQ outperform FIFO from 15 to 18%. This result agrees with the past studies (Holthaus and Rajendran 1997, Rajendran and Holthaus 1999). SRPT is slightly better than SPT and PT+WINQ, however for one of the job shop problem (abz5), SRPT underperforms FIFO rule. The table also indicates that recursive simulation based heuristics outperform FIFO rule from 14% to 23%, and the simulation based heuristics record better results than FIFO for all the shop problems. As expected, rDR1 and rDR2 give the best results since the heuristics use a multi-pass scheduling technique in addition to the recursive heuristic to improve on the performance measure. However the performance gain of the multi-pass scheduling procedures is not significant as compared to best result of the simpler recursive simulation heuristics, and the amount of improvement may not warrant the additional computation costs incurred.

All recursive simulation based heuristics record better minimum improvement on mean cycle time than dispatching rules. This minimum improvement can be considered as worst case performance guarantee for job shop problems. From the results, it seems that maximum improvement usually occur for job shop problems with large number of jobs. Apparently, larger number of jobs in the shop gives more choices to the machines and thus enable better job to be selected for the machines.

Table 1: Relative Performance of Mean Cycle Time for Dispatching Rules and Heuristics as Compared to FIFO

Relative Improvement to FIFO	**Mean (%)**	**Minimum (%)**	**Maximum (%)**	**Number Better (%)**
SPT	16.10	2.30 la19 (10x10)	38.85 ftp20 (20x5)	100.00
SRPT	17.74	-2.28 abz5 (10x10)	46.07 swv15 (50x10)	98.78
PT+WINQ	14.69	1.09 la36 (15x15)	39.04 swv12 (50x10)	100.00
rFIFO	13.62	3.57 la19 (10x10)	39.63 ftp20 (20x5)	100.00
rSPT	20.55	6.11 yn2 (20x20)	47.51 swv12 (50x10)	100.00
rSRPT	22.41	8.85 la39 (15x15)	49.82 swv14 (50x10)	100.00
rPT+WINQ	19.91	6.63 la24 (15x10)	44.21 swv15 (50x10)	100.00
rDR1	22.84	6.31 abz6 (10x10)	50.74 swv12 (50x10)	100.00
rDR2	23.20	6.69 abz6 (10x10)	50.74 swv12 (50x10)	100.00

Table 2 shows the results of relative performance of recursive heuristics relative to the corresponding dispatching rules. The results for the lower end improvement are encouraging as by simply applying recursive heuristic to simple dispatching rule FIFO, which is an average performer for mean cycle time, an improvement of 14% can be achieved. Even with the top performers of mean cycle time (SPT, SRPT and PT+WINQ), an improvement of 5 - 6% can still be attained relative their corresponding dispatching rules.

Table 2: Relative Performance of Mean Cycle Time for Recursive Heuristics Compared to the Corresponding Dispatching Rules

Relative Improvement to the Corresponding Dispatching Rule	**Mean (%)**	**Minimum (%)**	**Maximum (%)**	**Number Better (%)**
rFIFO	13.62	3.57 la19	39.63 ftp20 (20x5)	100.00
rSPT	5.35	0.80 la18	14.03 orb1 (10x10)	100.00
rSRPT	5.62	0.73 yn3 (20x20)	16.30 abz5 (10x10)	100.00
rPT+WINQ	6.24	0.50 yn2 (20x20)	12.38 swv06 (20x15)	100.00

Table 3 compares the top performers of recursive heuristics with tabu search procedures TB1 and TB2. The results show that the recursive heuristics are comparable to tabu search procedures. The improvement of TB2 over TB1 is relatively small, considering the much higher computation costs incurred (refer to Figure 2).

Table 3: Relative Performance of Mean Cycle Time for Top Recursive Heuristics and Tabu Search as Compared to FIFO

Relative Improvement to FIFO	Mean (%)	Minimum (%)	Maximum (%)	Number Better (%)
rSRPT	22.41	8.85 la39 (15x15)	49.82 swv14 (50x10)	100.00
rDR1	22.84	6.31 abz6 (10x10)	50.74 swv12 (50x10)	100.00
rDR2	23.20	6.69 abz6 (10x10)	50.74 swv12 (50x10)	100.00
TB1	20.99	5.60 la36 (15	46.93 swv15 (50x10)	100.00
TB2	22.24	5.79 la36 (15	47.85 swv15 (50x10)	100.00

Figure 2 presents the computation costs incurred for the 82 job shop problems for the dispatching rules and simulation heuristics. The execution speed of recursive simulation based heuristics rFIFO, rSPT, rSRPT and rPT_WINQ is about 15 times the corresponding dispatching rules. However, rDR1 and rDR2 record much higher computation costs, about 80 and 190 times respectively compared to the dispatching rules.

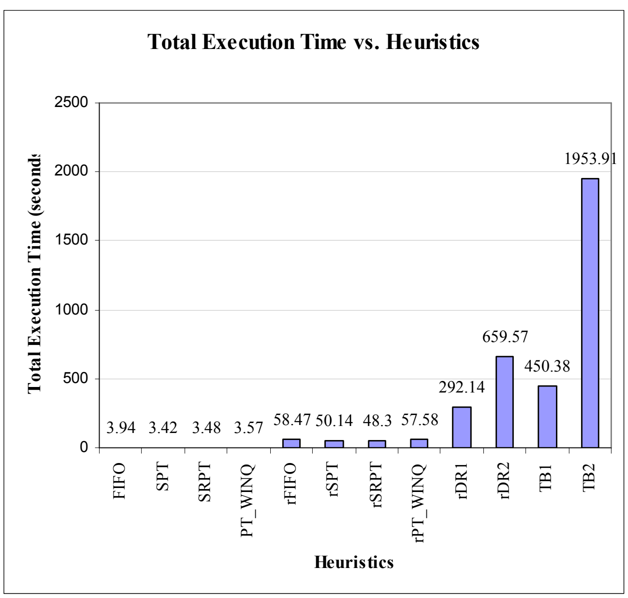

Figure 2: Total Execution Times of Various Heuristics and Procedures for the 82 Job Shop Problems

Depending on the performance requirement and computing constraints, we can thus determine the most appropriate dispatching rules or heuristics based on the results. For example, if we are interested in very fast approach to job shop scheduling, dispatching rule SPT is a good choice with worst performance guarantee. However, if we can compromise execution speed for the cycle time performance, rSRPT or even rDR1 can be better alternatives.

5 CONCLUSIONS AND FUTURE WORK

We have implemented a recursive simulation based scheduling approach to improve on the mean cycle time performance of dispatching rules. Dispatching rules are commonly known to be highly problem dependent in nature. By applying recursive simulation technique, we are able to increase the search space dimension of dispatching rules and to generate better solutions. The initial results show the recursive heuristic is comparable to other efficient procedures such as tabu search.

Our preliminary work indicates that the recursive simulation technique is intuitive and easily customizable for different dispatching rules and performance criteria by incorporating the rules and criteria within the recursive simulation framework. Multiple levels of recursive simulation runs can also be applied to enhance performance further if computation costs are not an issue.

We intend to test the recursive simulation based framework on a semiconductor manufacturing process by using commonly found dispatching rules in the semiconductor industry. One of the work we intend to pursue is to incorporate the simulation based approach into symbiotic simulation of semiconductor and test operation (Low et. al. 2005). Considering the fact that dispatching rules are often used in semiconductor manufacturing due to the complexity of the process, a few percentage improvements in performance could translate to huge savings in manufacturing costs.

Further work can also be done to improve the recursive simulation technique, in particular, the computation expense. The recursive heuristic can be restricted to only important decisions, and the time frame of each recursive run can be reduced. It is also possible to consider different scope and resolution for different decisions. Lower bound of each choice at a decision point can be estimated prior to the recursive runs to identify the most promising choices. Distributed and parallel simulation technique can also be applied to run multiple recursive runs concurrently.

REFERENCES

Adams, J., Balas, E. and Zawack, D. 1998. The Shifting Bottleneck Procedure for Job Shop Scheduling, *Management Science*, 34 (1): 391-401.

Armentano, V. A. and Scrich, C.R. 2000. Tabu Search for Minimizing Total Tardiness in a Job Shop, *Interna-*

tional Journal of Production Economics, 63 (1): 131-140.

Applegate, D. and Cook, W. 1991. A Computational Study of the Job Shop Scheduling Problem, *ORSA Journal on Computing*, 3 (2): 149-156.

Balas, E. and Vazacopoulos, A. 1998. Guided Local Search with Shifting Bottleneck for Job Shop Scheduling, *Management Science*, 44 (2): 262-275.

Blackstone, J. H., Philips, D. T. and Hogg, G. L. 1982. A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations, *International Journal of Production Research*, 20 (1): 27-45.

Blazewicz, J., Domschke, W. and Pesch, E. 1996. The Job Shop Scheduling Problem: Conventional and New Solution Techniques, *European Journal of Operational Research*, 93 (1): 1-33.

Brucker, P., Jurisch, B. and Sievers, B. 1994. A Branch and Bound Algorithm for the Job Shop Scheduling Problem, *Discrete Applied Mathematics*, 49 (1): 109-127.

Buffa, E. S., 1968. *Operations Management: Problems and Models*, New York: John Wiley & Sons. 454-460.

Chen, D., Turner, S. J., Cai, W. T., Gan, B. P. and Low, Y. H. 2004. Incremental HLA-based Distributed Simulation Cloning, In *Proceedings of the 2004 Winter Simulation Conference*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 386-394. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Conway, R. W. and Maxwell., W. L. 1962. Network Dispatching by Shortest Operation Discipline, *Operations Research*, 10 (1): 51-62.

Elvers, D. E. 1974. The Sensitivity of the Relative Effectiveness of Job Shop Dispatching Rules with Various Arrival Distributions, *AIIE Transactions*., 6 (1): 41-54.

Fisher, H. and Thompson G. L. 1963. Probabilistic Learning Combination of Local Job Shop Scheduling Rules, *Industrial Scheduling*, 225-251.

Ganesan, V. K., Sivakumar A. I., and Srinivasan G. 2004. Hierarchical Minimization of Completion Time Variance and Makespan in Jobshops, *Computers & Operations Research*, (In press.)

Gere, W. S., Jr. 1966. Heuristics in jobshop scheduling, *Management Science*, 13 (1): 167-175.

Gilmer, J. B. and Sullivan. F. J. 2000. Recursive Simulation to Aid Models of Decisionmaking, In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick. 958-963. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Gilmer, J. B. and Sullivan. F. J. 2003. The Use of Recursive Simulation to Support Decisionmaking, In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice. 1116-1121. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Gilmer, J. B. 2004. An Urban Terrain Abstraction to Support Decision Making Using Recursive Simulation," In *Proceedings of the 2004 Winter Simulation Conference*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 982-988. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Holthaus, O. and Rajendran, C. 1997. Efficient Dispatching Rules for Scheduling in a Job Shop, *International Journal of Production Economics*, 48 (1): 87-105.

Hopp, W. J. and Spearman. M. L. 1996. Factory Physics: Foundations of Manufacturing Systems, Irwin, USA: Times Mirror.

Jain. A. S. and Meeran. S. 1999. Deterministic Job Shop Scheduling: Past, Present and Future, *European Journal of Operational Research,* 113 (2): 390-434.

Jeong, K. C. and Kim, Y. D. 1998. A Real-time Scheduling Mechanism for a Flexible Manufacturing System: Using Simulation and Dispatching Rules, *International Journal of Production Research,* 36 (9): 2609-2626.

Law, A. M. and Kelton, W. D. 1991. *Simulation Modeling and Analysis,* New York: McGraw-Hill, 298-323.

Lawrence, S. 1984. Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques (Supplement), Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburg, 1984.

Lee, L. H., Tang, L. C. and Chan, S. C. 2001. Dispatching Heuristic for Wafer Fabrication, In *Proceedings of the 2001 Winter Simulation Conference,* e.d. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer. 1215-1219. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Low, Y. H., Lye, K. W., Lendermann, P., Turner, S. J., Leo, S. and Chin, R. 2005. An Agent-based Approach for Managing Symbiotic Simulation of Semiconductor Assembly and Test Operations, In *Proceedings of the 2005 International Conference on Autonomous Agent and Multiagent Systems*, Utrecht, The Netherlands.

Mazziotti, B. W., and Horne, Jr., R. E. 1997. Creating a Flexible, Simulation-based Finite Scheduling Tools, In *Proceedings of the 1997 Winter Simulation Conference.* ed. S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, 853-858. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Min, H. S. and Yih, Y. W. 2003. Development of a Real-time Multi-objective Scheduler for a Semiconductor Fabrication System, *International Journal of Production Research*, 41 (10): 2345-2364.

Morito, S. and Lee, K. H. 1997. Efficient Simulation / Optimization of Dispatching Priority with "Fake" Processing Time, In *Proceedings of the 1997 Winter Simulation Conference,* ed. S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson, 872-879. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Nanot, Y. R. 1963. An Experimental Investigation and Comparative Evaluation of Priority Disciplines in Job-shop Like Queuing Networks, Management Science Research Project, UCLA, Research report, No. 87.

Nowicki, E. and Smutnicki, C. 1996. A Fast Taboo Search Algorithm for the Job Shop Problem, *Management Science*, 42 (6): 797-813.

Rajendran, C. and Holthaus, O. 1999. A Comparative Study of Dispatching Rules in Dynamic Flowshops and Jobshops, *European Journal of Operational Research,* 116 (1): 156-170.

Sivakumar, A. I. 1999. Optimization of Cycle Time and Utilization in Semiconductor Test Manufacturing Using Simulation Based, On-line Near-real-time Scheduling System, In *Proceedings of the 1999 Winter Simulation Conference*, ed. P.A. Farington, H.B. Nembhard, D.T. Sturrok, and G.W. Evans, 727-735. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Storer, R. H., Wu, S. D. and Vaccari, R. 1992. New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling, *Management Science*, 38 (1): 1495-1509.

Yang, Y. and Chang, T. S. 1998. Multiobjective Scheduling for IC Sort and Test with a Simulation Testbed, *IEEE Transactions on Semiconductor Manufacturing*, 11 (1): 304-315.

Yoo, T. J., Kim, D. H. and Cho, H. B. 2004. A New Approach to Multi-pass Scheduling in Shop Floor Control, In *Proceedings of the 2004 Winter Simulation Conference*, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 1109-1114. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

CHIN SOON CHONG obtained his degree in Electrical and Electronics Engineering from the City University of London, UK. He joined Singapore Institute of Manufacturing Technology (SIMTech), a research institute in Singapore, ten years ago and is currently in the Manufacturing Planning and Scheduling Group of Manufacturing Information Technology Division. He obtained his Master of Engineering in Computer Integrated Manufacturing from Nanyang Technological University, Singapore. During these ten years, he has been involved in simulation, scheduling and optimization related projects in logistic and manufacturing IT domains. The industrial projects include cargo container operation and yard capacity planning simulation for a container port, dairy-printing process simulation for a printing company, manufacturing cycle-time modeling, scheduling and optimization for various MNCs. His current research interest includes intelligent and integrated simulation, planning, scheduling, optimization in the area of manufacturing, logistic, and supply chain. He can be reached via email at cschong@simtech.a-star.edu.sg.

MALCOLM YOKE HEAN LOW is a Research Engineer at the Singapore Institute of Manufacturing Technology. He received his Bachelor and Master of Applied Science in Computer Engineering from Nanyang Technological University, Singapore in 1997 and 1999 respectively, and a D.Phil. in Computer Science from Oxford University in 2002. His research interests include adaptive tuning and load-balancing for parallel and distributed simulations, and the application of multi-agent technology in supply chain logistics coordination. His email address is yhlow@simtech.a-star.edu.sg.

APPA IYER SIVAKUMAR (Sen-ior member IIE) is an Associate Professor in the School of Mechanical and Production Engineering at Nanyang Technological University (NTU), Singapore and a Fellow of Singapore Massachusetts Institute of Technology (MIT) Alliance (SMA). Prior to this he was at Singapore Institute of Manufacturing Technology, Singapore (SIMTech). His research interests are in the area of simulation-based opti-mization of manufacturing performance, supply chain, and dynamic schedule optimization. Prior to joining SIMTech in 1993, he held various management positions including technical manager and project manager for nine years at Lucas Systems and Engineering and Lucas Automotive, UK. He received a Bachelors of Engineering from Univer-sity of Bradford, UK and a PhD in Manufacturing Systems Engineering from University of Bradford, UK. He has been the technical chair and co-edited the proceedings of the 3rd and 4th International Conference on Computer Integrated Manufacturing (ICCIM ’95 and ICCIM’97), Singapore. His email and web addresses are msiva@ntu.edu.sg and www.ntu.edu.sg/home/MSiva/.

ROBERT GAY obtained his B. Eng, M. Eng and PhD degrees at the University of Sheffield, England, in 1965, 1967 and 1970 respectively. He was awarded the Grouped Scholarship in Engineering and Metallurgy by the University of Sheffield from 1967 to 1970. Since obtaining his PhD, he has been involved in Education and R&D working in institutions such as Singapore University (1972-1979), Rutherford and Appleton Laboratory (Eng-land, 1979-1982), NTU (1982-1995 and 1999-present) and Singapore Institute of Manufacturing Technology (1989-1999). He has also been actively involved in promoting innovation in Singapore through work in various commit-tees: Science Quiz (MOE), Science Centre Board, National CAD/CAM (NCB), Tan Kah Kee Young Inventors Award (TKK Foundation & NSTB), National IT Plan (NCB), Technopreneurship incubation center (ITE), Commercenet Singapore, Singapore Computer Society. Currently Profes-sor

Gay is in the School of EEE at NTU and also Director and CEO of the ASP Centre. He has more than a hundred publications in journals, conference proceedings and books. His email and web addresses are eklgay@ntu.edu.sg and www.ntu.edu.sg/eee/icis/cv/robertgay.html.