EFFICIENT STATE-INDEPENDENT SEQUENCING RULES FOR CERTAIN FLEXIBLE MANUFACTURING SYSTEMS

Arne Thesen

Department of Industrial Engineering
University of Wisconsin-Madison
Madison, WI 53706, U.S.A.

ABSTRACT

We introduce three state-independent methods for sequencing production of parts with random processing times on a simple production system such that productivity is maximized. The system of interest is one where parts of several different types are first processed on a common machine after which they proceed to separate, type-specific production lines. Referred to as *random*, *rotation*, and *most-behind* the rules are all based on easily computed estimates of the optimal product mix for the given part routings. Extensions of these rules are shown to give close to optimal performance in the case where blocking is present. The rules differs from conventional sequencing heuristics in that their performance is excellent over a wide range of system parameter values. The rules differ from optimal, state-dependent, scheduling rules in that no rule base or information about the current system states is needed.

1 INTRODUCTION

Consider the problem of determining the processing sequence of parts in a simple production system such that the number of parts produced per day is maximized. If all part routings and processing times are known in advance, then the consequences of all sequencing decisions can be predicted in advance, and the problem can be treated as a combinatorial optimization problem. In this case, a deterministic sequence of operations can be developed prior to the start of operations. On the other hand, if only the *distributions* of the processing times are known in advance, then optimal schedules cannot be developed in advance. Instead, state dependent scheduling rules leading to optimized expected performance can, under certain circumstances be developed. Based on assumptions of exponential service times and enumerable state spaces, these methods usually employ semi-Markov decision processes (SMDP, Howard, 1971) to develop specific scheduling decisions

for each individual system state. Typical applications can be found in Seidmann (1988), Yih and Thesen (1991), and Chen (1992). Efficient algorithms for solving SMDP problems can be found in Stidham and Weber (1993). Given the need for real time information and the need to accommodate a large and complex rule base, the resulting scheduling rules are often of limited practical value. However they can be of significant value as benchmarks for evaluating more implementable heuristic rules (see, for example, Seidmann and Tenenbaum, 1994).

Many simple scheduling rules are *static* in the sense that priorities are established prior to production, and remain unchanged over the life of the process. Common sequencing rules such as *Shortest Processing Time First* (*SPT*) and *Longest Processing Time First* (*LPT*) are static rules. Harrison and Wein (1990) introduced a static rule that performs well for a class of small systems using a small number of circulating pallets. Chen and Thesen (1995) show that the resulting performance is very good, but never optimal.

Given the difficulties in implementing state-dependent scheduling rules, the scheduling rules that we have encountered most frequently in practice in systems with dynamically changing workloads are still of the simple FIFO variety. In this paper we show that significant improvements in performance are possible if more carefully prepared state-independent scheduling rules are used.

Our ultimate goal is to develop systems with performance equal or close to the one that is attainable by optimized state depended scheduling rules, but without the need for any real time decision making at all. Towards this end, we introduce the following three different management procedures:

- *Random* (the next job is selected at random according to a pre-computed distribution)
- *Most Behind* (the next job is the one most behind its current production target)

- *Rotation* (the next job is the next entry in a pre-computed sequence)

Each of these rules employs pre-computed parameters. These parameters are based on the product mix resulting from a relatively simple linear program (LP) optimizing the production rates when the effects of queuing delays and blocking are ignored. We will see that all rules perform exceptionally well when large buffer sizes are allowed. Modifications of these rules designed to avoid delays due to blocking extend this excellent performance to cases with small buffer sizes.

This paper is organized as follows. The production system of interest is described in section two, as is a simple LP algorithm for estimating an upper bound of productivity and the corresponding product mix. The three state-independent scheduling procedures are introduced in section three, as are extensions to deal with the problem of blocking. An evaluation is presented in section four. A summary and conclusions are given in section five.

2 THE PROBLEM

The goal of this study is to develop new simple sequencing rules for a stochastic production system first studied by Seidmann (1988) (Figure 1). Briefly, the production system is defined as follows. Parts of n different types are first machined on a shared CNC, then they proceed to dedicated assembly lines (one for each part type). Processing times are exponentially distributed and buffer spaces are limited. Parts can be processed in any sequence on the CNC and there is an infinite supply of raw material. Our objective is to determine, in real time, a sequence of products to be produced at the CNC such that the overall production rate is maximized. A table summarizing the key features of this problem is given in Table 1.

2.1 Product Mix Estimates

We will refer to the *product mix* as the set (p) of percentages $(p_1, p_2, ..., p_i)$ describing the ratio between the output of individual parts and total production. For the class of systems studied here, total production (measured in parts per hour) is known if p and the utilization of any one machine are known. We will refer to the *optimal product mix* (P) as the product mix that leads to the highest possible production rate. It follows that any scheduling rule that gives 100% utilization of at least one machine while at the same time ensuring that the optimal product mix is maintained, will give the highest possible (e.g. optimal) production rate. The problem is

now reduced to one of finding the optimal product mix, and a method for ensuring 100 percent utilization of one machine.
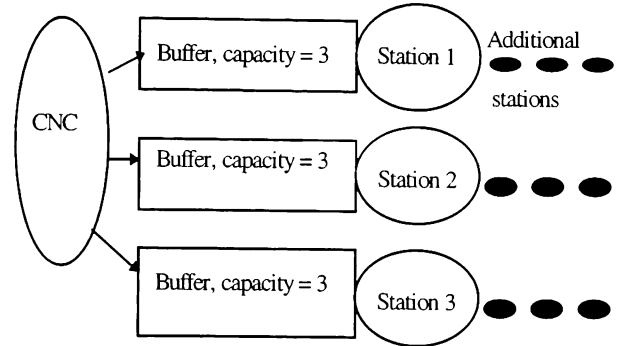

Figure 1: A FMS consisting of a CNC feeding parts of different types to three separate assembly lines

Table 1: Key elements of the problem of interest

Parts	Parts of n different types are produced. There is an unlimited supply of raw materials for all parts. All parts produced by the system can be sold.
Facilities	All parts are first processed at a single machining center. They then continue to separate assembly lines for each part type. The first station in each line is the bottleneck; hence subsequent stations need not be analyzed. Buffer capacities (b) in front of each line are identical.
Processing	Processing times are exponentially distributed random variables with the following parameters: m_i = Mean processing time at the machining center for parts of type i. A_i = Mean processing time at station 1 for parts of type i.
Control	*CNC*: Parts may be processed in any sequence. The identity of the part to be processed next is determined at the time processing starts. *Assembly lines*: Any part in the input buffer may be selected for processing.
Available Inform-ation	Three types of information are available: • Processing times (actual or expected) • System states (Buffer full/ not full, expected completion times) • Product mix (Target and/or actual)
Objective	Maximize parts produced per hour.

Table 2: The part routings used in this study

Rou ting	Mean Processing Times						Parts per Hr.	Upper bounds on throughput rates						
	Machining Center (min.)			Assembly Lines (min.)				Product mix (%)			Machine Utilization (%)			
	m_1	m_2	m_3	a_1	a_2	a_3		p_1	p_2	p_3	cnc	r_1	r_2	r_3
1	0.12	.24	.36	2.0	3.0	1.0	110	27	18	55	50	100	100	100
2	0.24	.48	.72	2.0	3.0	1.0	110	27	18	55	100	100	100	100
3	.375	.75	1.125	2.0	3.0	1.0	80	37	25	38	100	100	100	.50
4	0.6	1.2	1.8	2.0	3.0	1.0	60	50	37	17	100	100	100	17
5	1.00	2.0	3.00	2.0	3.0	1.0	45	67	33	0	100	100	75	0
6	.36	.24	0.12	2.0	3.0	1.0	110	27	18	55	50	100	100	100
7	.72	.48	0.24	2.0	3.0	1.0	110	27	18	55	76	100	100	.50
8	1.125	.75	.375	2.0	3.0	1.0	100	20	20	60	100	67	100	100
9	1.8	1.2	0.6	2.0	3.0	1.0	80	0	25	75	100	0	100	100
10	3.00	2.0	1.00	2.0	3.0	1.0	60	0	0	100	100	0	0	100

We first develop an estimator of P. Ignoring inefficiencies due to starvation and blocking caused by finite buffer sizes, random service times, and poor decisions, we formulate a linear program that maximizes the throughput rate for this system. The objective is to:

Maximize z = $\sum_1^n x_i$ (total output per hour)

Subj. to: $\sum_1^n m_i * x_i \ <= 60$ (Cell capacity)

$a_i * x_i \ <= 60 \quad i = 1,..,n$ (Line capacity)

Where:

x_i = parts of type i produced per hour
a_i = mean processing time at the first
       station of the ith assembly line (minutes)
m_i = mean processing time at the
       cell center for the ith part type (minutes)
n = Number of part types.

We define the *ideal product mix* is as the fraction of each type produced in the bound established above. Specifically, we define P_i as:

$$P_1 \ = \ \frac{x_i}{\sum_{j=1}^{n} x_j}$$

Many simplifying assumptions were made above, and the ideal product mix may not be identical to the optimal product mix. However, since the solution of the LP given above is an upper bound, any schedule that is consistent with the solution to this LP must be optimal. Hence, a schedule resulting in the .ideal product mix and in full utilization of at least one machine is optimal. We will see that this is easily

achieved when the buffers are large (i.e. no blocking). We will also see, for the case when blocking is present, that the three different scheduling rules using the ideal P_i's as targets dominate the performance of other well known scheduling heuristics.

2.2 Scenarios

We will study the systems performance when producing parts using ten different part routings (Table 2). The upper bounds on throughput rates given in this table were established by solving the linear program given in the previous section using the appropriate parameters. All part routings use the same mean times at the assembly lines. However, to observe a number of different bottleneck conditions, five different ratios between processing times at the CNC and the assembly lines were used. The first five and the last five routings use identical processing times at the CNC, and they differ only in that processing times for part types one and three were reversed (i.e. 1,2,3 vs. 3,2,1). This was done to negate any arbitrary benefit from a given sequence established by the relative magnitudes of the processing times. Observe that the CNC is a bottleneck for scenarios 2 through 5 and 8 through 10, and that all three lines are bottlenecks only for scenarios 1, 2, and 6. Scenarios 5, 9 and 10 have less than three parts types in the ideal product mix. Buffer allocations are assumed in all cases to be identical for all machines.

2.3 "Optimal" Decision Rules

Seidmann (1988) showed that stochastic scheduling systems of the type discussed here can be modeled as a Semi-Markov Decision Processes (SMDP). Chen (1992) used a three-phase implementation of the value iteration algorithm to find recommended state transitions for each decision state for several variations of the example given above. Unfortunately Chen found that

the state space increases exponentially with the number of buffer spaces, and he was unable to develop optimal schedules for problems significantly larger than the ones studied here. One useful result of Chen's research was the observation that the resulting optimal schedule can often be expressed in a simple decision table. He also showed that the rules developed this way can easily be extended to larger problems. The table used for scenarios one through four is shown in Table 3.

Table 3: Optimal state dependent decision rules for scenarios one through four.

CONDITIONS			ACTIONS		
Space Available In			Start Production of Part of		
Line A?	Line B?	Line C?	Type A	Type B	Type C
Y	I	I	Y		
N	Y	I		Y	
N	N	I			Y

3 DECISION RULES

Here we introduce three procedures for initiating production of parts of different types at the CNC such that a given fraction of parts of each type are produced. The *random* rule picks work pieces at random, using probabilities corresponding to the given product mix. The *rotation* rule uses the knowledge of the product mix to implement a repeating fixed sequence of parts to be produced. Finally, the *most behind* selects the next work piece as a part most under represented in the current mix. Note that these rules are not static since part priorities will change dynamically as production progresses.

3.1 State Independent Rules

Random. Under the random rule, we draw the next part at random according to the ideal product mix fractions (P_i). In other studies, a random rule has never been shown to be effective. In fact such rules are often included in studies as a neutral benchmark to determine the effectiveness of other rules. However, since the random rule used here includes information about the ideal product mix, it is quite effective. The expected throughput rate for a given set of P_i 's is readily computed (although we will not be doing so here). One advantage of this rule is that it needs no information

about the current state of the system. A disadvantage is that it takes no action to avoid full buffers. This leads to blocking when the system is operated with limited buffer spaces.

Rotation. Rotation sequences are constructed such that parts of the same type are evenly spaced in the sequence, and such that the ratio between the different parts in the sequence is identical to the given product mix. For example the sequence (1, 2, 1, 1, 2, 1, 3,...) implements a rotation schedule for the 50 %, 33 %, 17 % ideal part mix for scenario 4. The rotation schedules used for all scenarios are shown in Table 4. Again, the expected throughput rate for a given rotation schedule is readily computed. By placing a given part type at even intervals in the sequence, the assembly line buffer for that part is given time to clear before a new part of the same type is produced. Simulation experiments show that this feature causes this rule to perform better that the random rule in avoiding blocking when buffer capacities are limited. A disadvantage of this rule is the fact that (short) rotation sequences are difficult or impossible to develop for many product mixes.

Table 4: Rotation schedules

Cases	Product mix	Production sequence
1,2,6,7	27,18,55	3, 1, 3, 2, 3, 1, 3, 2, 3, 1, 3, ...
3	37,25,38	1, 3, 2, 1, 3, 1, 3, 2, ...
4	50, 33, 17	1, 2, 1, 2, 1, 3, ...
5	67, 33, 0	1, 1, 2, ...
8	20, 20, 60	3, 2, 3, 1, 3, ...
9	0, 25, 75	3, 3, 3, 2
10	0, 0, 100	3

Most behind. Under this rule, we multiply the present total output by the ideal product mix probabilities to obtain the current production target for each part type. This value is compared to the actual production. The most under-produced part is selected for production. The performance of this rule is quite similar to the one observed for the Rotation rule. However, sequences can be developed with equal ease for any product mix. A dis-advantage of this rule is that it may be difficult to collect the required data in real time. Presently, we are unable to estimate the expected throughput rate for a given rotation schedule without using simulation.

Table 5. Some state-independent rules

Rule	Description	Real Time Information Needed	Avoids full buffers	Remarks
Random	Pick next part at random, in	None	No	

	accordance with the ideal product mix			
Rotation	Pick part form a pre-computed fixed sequence.	None	Yes	Rotation sequences difficult to develop
Most Behind	Pick part most under represented in current product mix	Current product mix	Yes	Probably good for smaller buffers

3.2 State-Dependent Extensions

The simple sequencing rules introduced above perform well when the assembly lines have large buffer capacities. However, frequent delays due to blocking occur when buffer spaces are limited. Here we will discuss ways to avoid this problem while retaining the ideal product mix generated by these rules.

Unequal buffer allocations. Delays due to blocking can be substantially reduced by reallocating limited buffer spaces from poorly utilized buffers to more highly utilized ones. Thesen and Chen (1996) shows that optimal scheduling rules change when this is done, and they propose an algorithm for simultaneously establishing optimal buffer allocations and scheduling rules. Heuristic scheduling rules such as *Probabilistic Shortest Queue* (*PSQ*, Yao and Buzacott, 1984) and *Fastest Shortest Queue* (*FSQ*, Seidmann and Tenenbaum, 1994) that use information about currently available buffer capacities work very well when limited buffer spaces are optimally allocated.

Blocking avoidance. The simplest way to avoid blocking is not to start processing of a part at the CNC if it will be going to a full buffer. This may be achieved by excluding parts going to full buffers from consideration for processing at the CNC when the next work piece is selected. A problem with this strategy is that the resulting product mix may deviate from the ideal one for some rules. In fact, only the *most-behind* rule will self-correct for the distortions introduced by this scheme.

Anticipation. It is difficult to avoid blocking when all buffers are full. One efficient strategy is to select the part with the longest possible processing time. Also, process completion times can very often be determined with great accuracy when processing starts. Using this knowledge, the CNC scheduler may be able to select a part for processing, knowing that there will be room in the corresponding input buffer when processing at the CNC is completed. However, the resulting product mix may again be distorted.

4 EVALUATION

In this section we report on experiments designed to evaluate the decision rules discussed in the previous section. For reference purposes, the conventional *SPT* rule is also evaluated. Simulation using ProModel for Windows (Baird and Leary, 1994) was used to estimate the throughput rates resulting from the use of different decision rules. The batch means technique (10 batches, 10,000 parts per batch) was used to estimate a confidence interval for the throughput rate for each rule under each scenario. All confidence intervals have half widths less that 5% of the mean. Only the resulting means are shown in this report. The optimal throughput rates were obtained using a value iteration based SMDP algorithm due to Chen (1992).

4.1 State-Independent Rules

We first evaluated performance under the conditions of "weak blocking". In this case we assumed that 30 buffer spaces were available in front of each line. The results for scenarios one through five are shown in Figure 2. We see that performance within a few percent of optimal was observed in all cases. However, we also see that the conventional SPT rule performed rather poorly. The results for scenarios six through ten are quite similar.

The performance observed for these rules under scenarios one through five with three buffer spaces in front of each line is shown in Figure 3. We see that the *most behind* and *rotation* rules resulted in relatively good performance, while the random rule performed less well. In all cases did the optimal rule result in the best performance. An analysis of the simulation output indicates that the reason for the inferior performance of the random rule is that blocking is more likely to occur as full buffers are not avoided. The results are similar for scenarios 6 - 10.

4.2 State-Dependent Rules

Figure 4 gives the observed performance for all ten scenarios when the rules are modified to implement the state-dependent suggestions given for blocking avoidance and anticipation in the previous section. The

difference in performance between the modified rules and the optimal rules is not statistically significant at the 5% level. A separate test was run to determine if performance improved when information about buffer availability at the *end* of processing at the CNC was used in place of knowledge about buffer availability at the *start* of processing. No significant improvement was found.
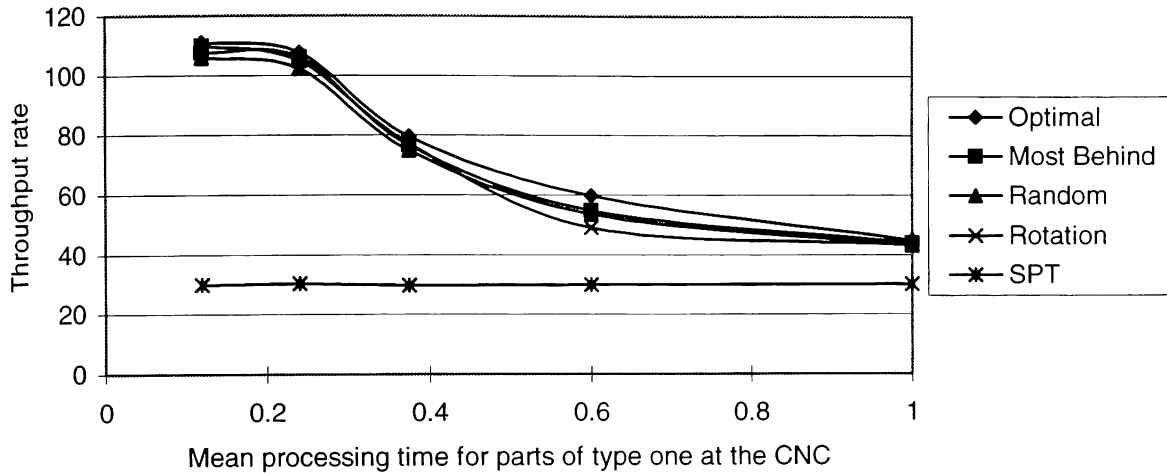

Figure 2: Performance of the simple scheduling rules for scenarios one through five under the condition of weak blocking. Thirty buffer spaces are available at each of the assembly stations
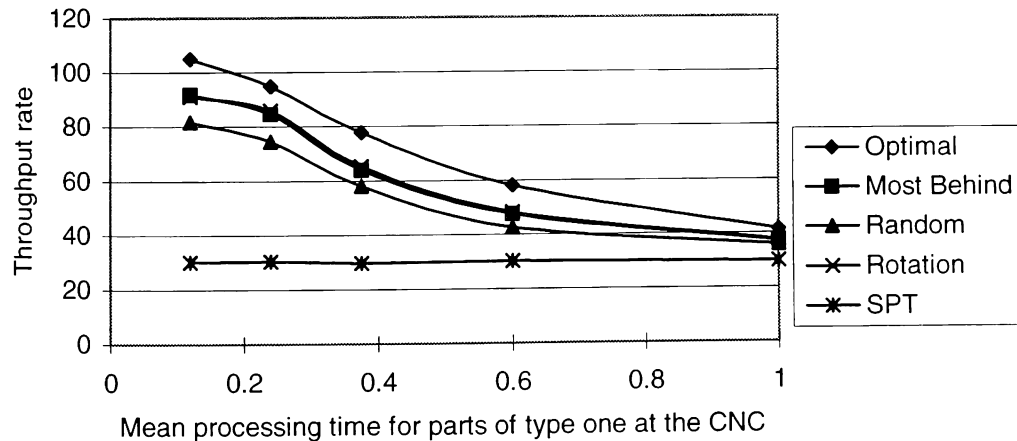

Figure 3: Performance of the simple scheduling rules for scenarios one through five under the condition of strong blocking. Three buffer spaces are available at each of the assembly stations
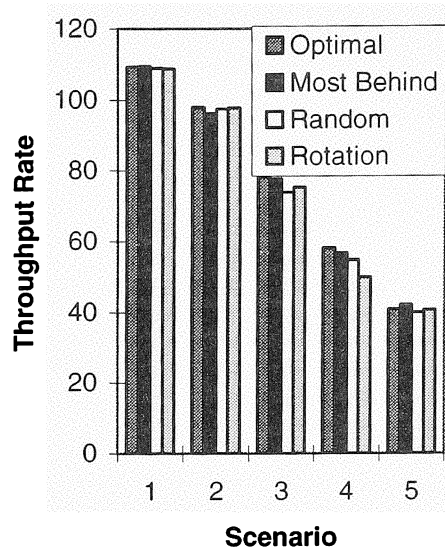
Figure 4: Throughput rate for state dependent rules. Three buffer spaces at each of the assembly stations

5 CONCLUSION

We have introduced three different sequencing rules all of which exploit information about an optimal product mix that is easily computed using information about part routings. The rules were shown to give good to excellent performance in a predictable manner for a wide range of scenarios. The choice of a rule for a given situation depends on the availability of buffer spaces, and on the information available to the scheduler. If the system has large buffer spaces and if no real-time information is available about buffer populations, then the random rule should work well. When buffer spaces are limited and real time information is available, then the *most-behind* rule should give close to optimal performance.

REFERENCES

Baird, S.P., and J.J. Leavy. 1994. Simulation Modeling Using Promodel for Windows. In *Proceedings of the 1994 Winter Simulation Conference*, eds. Jefferey D. Tew, Mani S. Manivannan, Deborah A. Sadowski, and Andrewe F. Seila, 527-532.

Chen, J.C., 1992, *State Dependent Scheduling for Manufacturing Systems*, Ph.D. Dissertation. Department of Industrial Engineering, University of Wisconsin-Madison.

Chen, L-Y., and A. Thesen. 1995. Performance Analysis of A Two-Station Closed Queueing Network Using Static and Optimal Sequencing Policies. *Technical Report 95-5*, Industrial Engineering, Univ. of Wisconsin at Madison.

Harrison, J. M., and L. M. Wein, 1990. Scheduling Network of Queues: Heavy Traffic Analysis of a Two-Station Closed Network. *Operations Research* **38**, 1052-1064.

Howard, R.A., 1971, *Dynamic Probabilistic Systems. Volume II: Semi-Markov and Decision Processes*, John Wiley & Sons, New York, New York.

Seidmann, A., 1988, "Optimal Dynamic Routing in Flexible Manufacturing Systems with Limited Buffers", *Annals of Operations Research*, 15, 291-311.

Seidmann, A., and A. Tenenbaum, 1994. Throughput Maximization on Flexible Manufacturing Systems. *IIE Transactions*, Vol. 26, No. 1, 90-100.

Stidham, S., and R. Weber,1993, "A Survey of Markov Decision Models for Control of Networks of Queues", *Queueing Systems*, 13, 291-314

Thesen, A., and J.T. Chen , 1996, "Jointly Optimized Buffer Allocation and Sequencing Rules for a FMS", *Proceedings of IASTED Conference on Modelling, Simulation and Optimization*", Gold Coast, Australia, May 6.

Yao, D.D., and J.A. Buzacott, 1984, "Modeling a Class of State-Dependent Routing in FMSs", *Annals of OR*, 3, 153-168

Yih, Y., and A. Thesen, 1991,. "Semi-Markov Decision Models for Real-Time Scheduling". *International Journal for Production Research*, Vol. 29, No. 11, 2331-2346.

AUTHOR BIOGRAPHY

ARNE THESEN is the Robert A. Ratner Professor of Industrial Engineering at the University of Wisconsin-Madison. He holds a MSIE and Ph.D. from the University of Illinois Urbana-Champaign, and he is a member of INFORMS and a Senior Member of IIE. He has authored or co-authored three textbooks in the areas of simulation and operations research, and he has published over 40 papers in the same areas. A former department chair, his current interests are in short term scheduling and heuristic optimization methods.