ROBOT SCHEDULING IN A CIRCUIT BOARD PRODUCTION LINE: A HYBRID OR/ANN APPROACH

YUEHWERN YIH
School of Industrial Engineering, Purdue University, West Lafayette, IN 47907

TING-PENG LIANG AND HERBERT MOSKOWITZ
Krannert School of Management, Purdue University, West Lafayette, IN 47907

A hybrid method that combines human intelligence, an optimization technique (semi-Markov decision model) and an artificial neural network to solve real-time scheduling problems is proposed. The proposed method consists of three phases: data collection, optimization, and generalization. The testbed of this approach is the robot scheduling problem in a circuit board production line where one overhead robot is used to transport jobs through a line of sequential chemical process tanks. Because chemical processes are involved in this production system, any mistiming or misplacing will result in defective jobs. The proposed hybrid system performs better than the human scheduler from whom the models were formulated, both in terms of productivity and quality.

■ Due to the dynamic nature of current manufacturing environments, real-time scheduling has become a priority of research focus. The goal of real-time scheduling is to make the best scheduling decisions to meet the production demand based on the limited information currently available. The concept of real-time scheduling was first introduced in developing computer operating systems in the late 1960s. Because a computer connects to a number of terminals that may send jobs at any time they are in use, it is extremely difficult for the operating system to plan and schedule the jobs in advance. Scheduling must be done on a real-time basis. In the manufacturing domain, many problems have similar characteristics. For example, when a material handling robot is used to move in-process products between stations, real-time scheduling is useful when uncertainties in processing time and machine failure exist.

Traditionally, interactive scheduling that requires human experts to control the process was suggested for handling the real-time scheduling problem (Godin [3], Haider *et al.* [4], Hodgson and McDonald [5], Hurrison [7]). This poses several difficulties. Human experts often take a considerably long time to train. Moreover, their performance may be inconsistent and/or biased. Furthermore, employee turnover often causes major problems.

Recently, a strong desire for automation has motivated research on automated scheduling. An automated scheduling system consists of sensing devices, a scheduler, and control devices. The sensing devices collect data about the current state of the system. The scheduler analyzes the collected information and decides the next course of action. The decision is relayed to proper control devices for execution. The automated scheduler is the core of the system.

There are two existing approaches to designing an automated scheduler. One uses optimization techniques such as integer programming (Sharpiro and Nuttle [16]) or a semi-Markov model (Yih [22]); and the other uses an expert systems approach that incorporates heuristic rules into computer systems to solve the real-time scheduling problem (Fox and Smith [2]; Kusiak [8]; MacFarland and Grant [11]; Ow and Smith [13]; Thesen and Lei [18], [19]; Yih [21]). Because of the complexity of the problem, however, each of these approaches has certain limitations. The optimization approach needs to have complete knowledge of the problem states and often relies on some strong assumptions to control complexity, which reduces its applicability to real-world problems. The expert systems approach depends heavily on the quality of the rules in the knowledge base. Acquiring high quality knowledge, however, is difficult. Human experts may be either unwilling or unable to articulate their knowledge or, in some cases, experts are simply unavailable.

In this paper, a hybrid approach to automating real-time scheduling is proposed. This method consists of three phases: data collection, optimization, and generalization. A simulator is developed to collect data from human schedulers during the problem-solving process. The scheduling problem is modeled as a semi-Markov decision process using the collected data and an "optimal policy" is derived through the policy iteration method. An artificial neural network (ANN) is developed based on the "optimal policy" to form an automated scheduling system.

The optimized rules were then evaluated in a real-time scheduling environment. The results indicated that rules derived from the semi-Markov process yielded better performance than the actual scheduling decisions.

A limitation of the semi-Markov approach is that the optimized rules often cover only a subset of the whole state space because the transition probabilities collected from the expert data usually cover only a subset of the state space. As a result, when the rules are used on-line, the system becomes stalled every time a state not covered by the original data appears. Therefore, the data-based semi-Markov model alone does not provide the desired full automation.

Expert Systems

In addition to mathematical programming and semi-Markov approaches, expert systems that use heuristics have also been proposed for robot scheduling. For example, Lei and Thesen conducted simulation studies to examine the performance of several heuristic dispatching rules in different system states. They found certain rules to be better than others and the expert system approach to be very promising in this problem domain (Lei [10]; Thesen and Lei [18], [19]).

One major bottleneck in applying expert systems in general is the difficulty in obtaining reliable knowledge/expertise for decision making. Knowledge can be obtained by interviewing human experts and recording the rules these experts employ in scheduling. The problem is that human experts are often unable or unwilling to express their knowledge accurately and/or completely. Recent literature in knowledge acquisition has proposed the use of machine learning techniques to bypass human experts and automatically induce rules from expert's decisions (Quinlan [14]; Liang [9]; Yih [21]). This approach suggests that experts' decisions coupled with their system state descriptions be collected as training cases. These training cases can then be fed into inductive learning algorithms to derive rules that form the knowledge base of expert systems. Existing literature has reported promising results using this approach. For example, Yih [21] developed the Trace-Driven Knowledge Acquisition (TDKA) method to learn the scheduling rules from the "traces" (i.e., training cases) collected from expert schedulers, and found that the induced rules performed better than the human schedulers.

The inductive learning approach to knowledge acquisition also is not without its limitations. Because it produces rules from the training data, the quality of the training cases has an enormous effect on the reliability and performance of the induced rules. Most inductive learning algorithms have problems when "noisy" data exist in the training cases. Rules derived from noisy data are often error-prone and may result in poor system performance. Unfortunately, few generally applicable mechanisms for removing noise from the training data are available.

A Hybrid Approach

We now present a hybrid approach to the robot scheduling problem that combines semi-Markov decision models with ANNs. Semi-Markov decision models will be used to optimize the throughput based on the training cases collected from the simulation and ANN will then be applied to construct a scheduling model that covers the entire state space for on-line real-time scheduling. In the following sections, we outline the three major components of our integrated hybrid method: simulation, semi-Markov optimization, and ANN modeling.

Data Collection

The first step in the hybrid approach is to collect training data for modeling. In real-time scheduling, states are dependent upon the previous state and the decision made. Each decision will have a chain effect on its following states. Therefore, it is difficult for human schedulers to anticipate a state and explain the basis for their decisions over all possible states. In addition, the effect of a poor decision is difficult to assess because of its interactions with previous and subsequent decisions. Under these circumstances, the best approach for collecting training data is to have human schedulers make decisions in a simulated or real production line rather than collect information from interviews, where the computer captures the states and the corresponding decisions. In this way, their knowledge and expertise are documented in a series of decision making episodes.

A major advantage of using simulation is that it allows human schedulers to express their knowledge in a non-intrusive environment. Instead of explaining what they know, human schedulers solve the scheduling problem in real time and the computer captures their decisions for subsequent analysis. This can also be done by observing decisions unobtrusively in an actual production line wherever possible.

Improvement Procedure—Semi-Markov Decision Processes

Because human schedulers may exhibit certain biases or inconsistencies, the training data collected from schedulers may contain noise as well as errors. The semi-Markov technique serves as an optimization-based noise filter for heuristic-based AI methods. It finds the best decision for each state visited by the scheduler and removes the poor choices. This increases the consistency of the training data for neural network modeling.

The size of the state space in a real-world problem is often too large to obtain an analytical solution. Moreover, it is very difficult to calculate the transition probabilities for constructing a semi-Markov model. To mitigate these problems, the scheduling data collected in the first step are used to estimate the transition probabilities for the semi-Markov model formulation and reduce the state space to the one whose states have been visited by the human scheduler. Over a long period, the proportion of time spent in the states not visited by the schedulers could be very negligible. An iterative algorithm (Howard [6]) for solving the semi-Markov model is applied to derive the optimal policy. The detailed modeling processes could be found in Yih and Thesen [23].

The optimal policy contains the best decision for each state. The states included in the optimal policy, however, only cover those visited by the human scheduler and are by no means necessarily complete. Therefore, an AI method that can generalize the subset to the entire state space is necessary to construct an automated scheduler.

Generalization Process—ANN

ANN is a machine learning method that has recently gained popularity because of its analogy to the neural structure of the human brain. In fact, ANN is a model building technique that constructs non-linear decision models from training cases. The literature has indicated its promise as a modeling tool and ANNs can theoretically represent models of any functional form (White [20]). Successful applications of neural networks have been reported in engineering design and other problem domains (e.g., Chryssolouris, *et al.* [1]; Rauch [15]; Shepanski and Macy [17]).

There are many different ways to train a neural network. One of the most popular training algorithms is called backpropagation. It is basically a gradient descent process that randomly generates an initial set of weights and then continuously adjusts them to reduce the total sum of square errors to a satisfactory level (see e.g., McClelland and Rumelhart [12]). The output of a neuron j in an ANN model, O_j, is a function of its inputs, I_i, and the weights associated with all connections between neurons i and j, ω_{ij}. The most popular heuristics used for backpropagation-based learning are called delta rules, which estimate connection weights at time $t+1$, $\omega_{ij}(t+1)$, from weights at time t, $\omega_{ij}(t)$ (Figure 2).

In summary, our proposed hybrid approach to develop an automated robot scheduling system includes several steps as illustrated in Figure 3. First, human scheduling decisions are collected by simulation or other means. These data are used to estimate the state transition probabilities. Policy iteration procedure is then applied to determine the optimal policies for the states shown in the training cases. These optimal policies are a subset of the complete optimal decision set. Finally, the subset of optimal policies are used to train an ANN model. The ANN model generalizes the subset of optimal policies to cover the entire state space and can be installed on line for automated scheduling. This hybrid approach overcomes the drawback of using a semi-Markov model or ANN method *per se* and should achieve superior performance in real-time scheduling.

Experiment

The major difference between the robot (real-time) scheduling problem and other flow shop scheduling problems is that the manufacturing process involved is

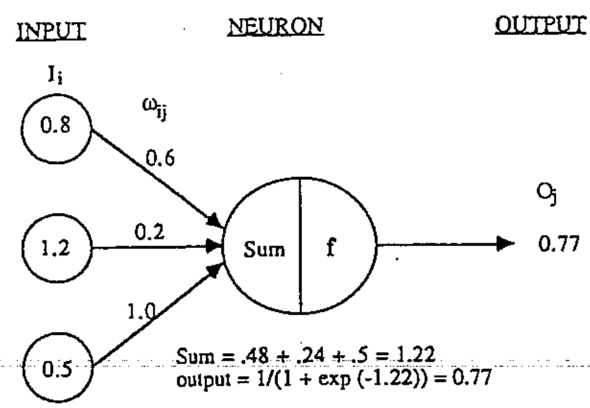

Figure 2. Calculations in a neuron

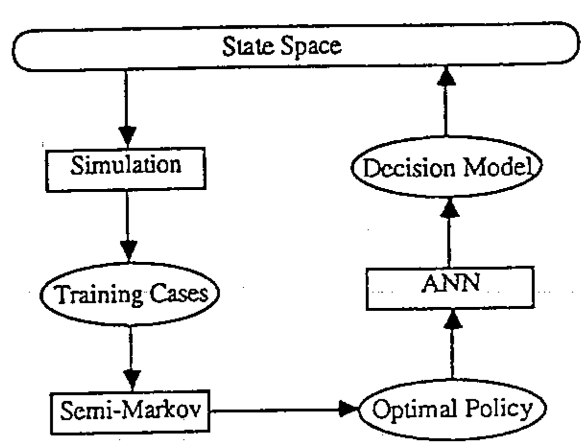

Figure 3. The hybrid approach

a chemical process, rather than a material removal process. Once a process is completed, the job has to be removed from the tank (workstation) to the next one immediately before it becomes "spoiled." If the job cannot be removed on time (because the robot is not available or the next tank is busy), it will become defective. Therefore, it is critical to determine when a new job should be entered into the system and which job to serve first during the process. If we assume that the robot movement time is negligible, the earliest time (ET) to enter a new job into the system to avoid a basic tank conflict can be computed through the procedure described in Appendix A. A job that enters at any time earlier than ET will definitely result in a basic tank conflict and become defective. Of course, a job that enters the system after ET cannot be guaranteed to be good, because we neglect the robot travel time. Meanwhile, the dispatching decisions during the process also affect the outcome. However, ET appears to be the best aid we can provide to the user, although we have considered other information aids as well, such as processing times for each job.

Method

To collect data from human subjects, an interactive simulation program describing the system of interest was developed, called *simulator*. It simulates the process in real time and provides information through the user-interface. The subject makes decisions using the available information and controls the system through the user-interface. The system status as well as its associated decision will be recorded for later analysis. Prior to the experiment, the subjects are first familiarized with the scheduling problem as well the *simulator* in the training process.

Each subject spent about three hours to understand the scheduling problem and learn to operate the simulation program. Four simulation runs were performed by each subject. Each simulation run took about two hours. After the optimization process, it took twenty hours to train the ANN. Because of the amount of time involved in the data collection process, two subjects participated in the experiment. To assure adequate data for analysis, each subject ran the simulation four times, yielding a total of eight runs. The performance measures of interest included product quality and throughput rate. *Product quality* is defined as the percent of non-defective, i.e.,

$$\text{Product Quality} = \text{No. of good jobs/Total no. of jobs processed.}$$

To discount defective products, *throughput* was defined as the weighted number of good jobs produced, i.e.,

$$\text{Throughput} = \text{No. of good jobs produced} * \text{Product quality.}$$

Each set of collected data was used to formulate a semi-Markov decision process and was optimized through the policy iteration algorithm (Howard [6]). In our semi-Markov decision model, the state definition can be stated as follows:

$$\text{State} = (T[0], T[1], T[2], T[3], T[4], T[5], CL, F1, F2)$$

Where

T[0] = 1 if a job is available in the input buffer without a basic tank conflict; 0 otherwise.

T[i] = the rank order of the remaining processing time in tank i ($i=1,2,\ldots,5$);
0 if tank i is empty;
-1 if the job in tank i is defective.

CL = the robot location.

F1 = 1 if the remaining processing time of the tank with rank number one is less than a threshold value;
0 otherwise.

F2 = 1 if the remaining processing time of the tank with rank number two is less than a threshold value;
0 otherwise.

For F1 and F2, the threshold value is the time needed for the robot to travel from the input buffer to the output buffer. A decision is defined as the number of the tank where the job will be removed next, that is, 0 for entering a new job to tank 1, 1 for moving the job in tank 1 to tank 2, and so forth. The reward associated with each decision is defined as the incremental value in the output buffer: +1 for a good throughput, -1 for a defective job, and 0 otherwise. Transition probabilities can be estimated by (Yih and Thesen [23]).

$$P_{ij}^{k} \approx \frac{\text{frequency count of transition from } i \text{ to } j \text{ under alternative } k}{\text{total frequency count of visiting state } i \text{ under alternative } k}.$$

Following are two illustrations from a derived optimal policy.

State	Decision
(1 0 3 0 2 1 4 1 0)	5
(1 0 2 0 1 0 6 1 0)	4

The derived optimal policy is then used as the training data for an ANN. In the neural network model, there exists three layers: an input layer, hidden layer, and output layer. The nodes between the input layer and hidden layer are fully connected, as are those between the hidden layer and the output layer. The input layer in our ANN contained 9 nodes, each of which corresponded to one attribute in the state definition. The output layer had 6 nodes which represented six possible decisions in the system. Twenty nodes were selected for the hidden layer after systematically experimenting with different numbers of hidden nodes based upon the performance of the resulting ANN. The ANN was trained for twenty hours on a 486 IBM-compatible personal computer which was determined by observing that the level of errors of the ANN was reasonably low and the decrement in error rate was approaching zero. The resulting network model was then evaluated in the real-time robot scheduling problem.

Results

Using common random number streams in the simulation runs, the resulting automated scheduling system from the hybrid method exhibited better performance than the human schedulers. For example, the throughput average over both subjects was increased from 210.89 for the human to 243.49 (t=2.634, $p < 0.05$) (Table 1). From Table 1 and Figure 4, we note that on average subject #2 performed better than subject #1 by 44% in terms of throughput. Interestingly, the throughput incremental gain of the hybrid method over the human scheduler from the data of subject #1 was 20.28%, while the performance improvement increment from the data of subject #2 was 12.12%. The hybrid method thus reduced the difference in throughput performance between the two subjects by 22.14%.

Also, as shown in Table 1 and Figure 5, average product quality was improved from 88.76% to 93.33% on average. Similar to the throughput measurement, the average quality improvement of the hybrid method for subject #1 (6.78%) was higher than subject #2 (2.28%). The difference in quality performance between the two subjects was thus reduced by 60.65%.

The performance of the hybrid scheduling system is clearly dependent upon the quality of the collected data from the human scheduler. However, the experimental results showed that the hybrid method is able to improve the system throughput and quality of human schedulers regardless of their level of expertise. Of course, the more expert the human scheduler is, the better the resulting scheduling system performs. However, the relative improvement of the hybrid approach appears

Table 1. Comparison of Human Scheduler and the Hybrid Method

Throughput

No.	Subject	Human Performance	Human Avg.	Hybrid Performance	Hybrid Avg.
1	#1 (1)	185.4		180.4	
2	#1 (2)	146.3		248.5	
3	#1 (3)	174.4		185.5	
4	#1 (4)	184.9	172.75	216.4	207.78
5	#2 (1)	240.0		269.3	
6	#2 (2)	253.8		249.6	
7	#2 (3)	238.8		282.0	
8	#2 (4)	263.5	249.02	315.9	279.20
	Avg.	210.89		243.49	

MEAN DIFFERENCE = 32.600
SD DIFFERENCE = 35.006
T=2.634 DF=7 PROB=.034

Product Quality

No.	Subject	Human Performance	Human Avg.	Hybrid Performance	Hybrid Avg.
1	#1 (1)	0.889		0.997	
2	#1 (2)	0.791		0.959	
3	#1 (3)	0.866		0.821	
4	#1 (4)	0.888	0.8585	0.914	0.9228
5	#2 (1)	0.916		0.973	
6	#2 (2)	0.923		0.925	
7	#2 (3)	0.894		0.912	
8	#2 (4)	0.934	0.9168	0.965	0.9438
	Avg.	0.8876		0.9332	

MEAN DIFFERENCE = 0.046
SD DIFFERENCE = 0.066
T=1.956 DF=7 PROB=.091

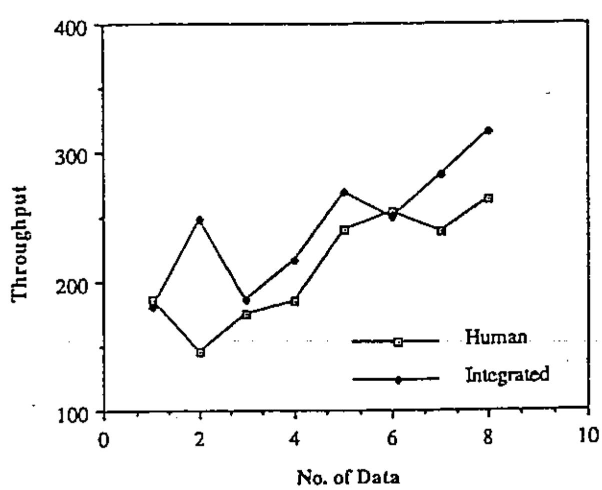

Figure 4. Throughput of human subjects vs. hybrid method

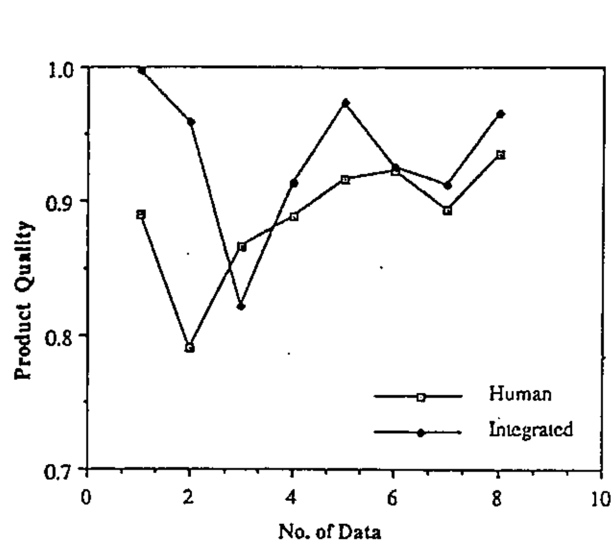

Figure 5. Product quality of human subjects vs. hybrid method

greater for more naive (less expert) schedulers. Indeed, the data collected from non-expert operators may benefit more performance-wise from the use of the hybrid method than those from expert operators.

Concluding Remarks

A hybrid approach for automated robot scheduling in circuit board production lines was proposed and the feasibility of such an approach demonstrated. The experimental study showed that the proposed method is able to effectively automate the decision-making process in real-time scheduling tasks and yields better performance than human schedulers. The significance of this research is that it demonstrates the feasibility and promise of integrating OR and AI techniques in real-time scheduling. Of further significance is the fact that our proposed hybrid approach is even relatively more effective in improving scheduling performance of decisions obtained from schedulers who are not necessarily experts in their operating domain.

Acknowledgments

This research was supported in part by grants from the Purdue Research Foundation, and the Center for Management of Manufacturing Enterprises.

REFERENCES

[1] Chryssolouris, G., Lee, M., Pierce, J., and Domroese, M., "Use of Neural Networks for the Design of Manufacturing Systems," *Manufacturing Review*, 3:3, 187-194 (1990).

[2] Fox, M. S., and Smith, S. F., "ISIS—A Knowledge Based System for Factory Scheduling," *Expert Systems Journal*, 1, 1, 25-49 (1984).

[3] Godin, V. B., "Interactive Scheduling: Historical Survey and State of the Art," *IIE Transactions*, 10, 3, 331-337 (1978).

[4] Haider, S. W., Moodie, C. L., and Buck, J. R., "An Investigation of the Advantages of Using a Man-Computer Interactive Scheduling Methodology for Job Shops," *International Journal of Production Research*, 19, 4, 381-392 (1981).

[5] Hodgson, T. J., and McDonald, G., "Interactive Scheduling of a General Flowshop," *Interfaces*, 11, 83 (1981).

[6] Howard, R. A., *Dynamic Probabilistic Systems, Volume II: Semi-Markov and Decision Processes*, John Wiley & Sons (1971).

[7] Hurrison, R. D., "Investigation of Visual Interactive Simulation Methods Using the Job-Shop Scheduling Problem," *Journal of the Operations Research Society*, 29, 1085 (1978).

[8] Kusiak, A., "Designing Expert Systems For Scheduling Of Automated Manufacturing," *Industrial Engineering*, 42-46 (July 1987).

[9] Liang, T. P., "A Composite Approach to Automated Induction of Knowledge for Expert Systems Design," *Management Science*, forthcoming (1991).

[10] Lei, L., "A State Dependent Approach to Supervisory Control of Robots," Ph.D. dissertation, University of Wisconsin-Madison (1988).

[11] MacFarland, D. G., and Grant, F. H., "Artificial Intelligence: Advances In Sequencing and Scheduling," *1987 IIE Integrated Systems Conference Proceedings* (1987).

[12] McClelland, J. L., and Rumelhart, D. E., *Explorations in Parallel Distributive Processing*, The MIT Press, Cambridge, MA (1988).

[13] Ow, P. S., and Smith, S. F., "Towards an Opportunistic Scheduling System," *Proceedings of the 19th Annual HICSS Conference*, Honolulu, Hawaii (January 7-10, 1986).

[14] Quinlan, J. R., "Induction of Decision Trees," *Machine Learning*, 1:1, 81-106 (1986).

[15] Rauch, H. E., "Neural Network for Routing Communication Traffic," *IEEE Control Systems Magazine*, Vol. 8, 26-31 (1988).

[16] Sharpiro, G. W., and Nuttle, H. L. W., "Hoist Scheduling For A PCB Electroplating Facility," *IIE Transactions*, 20, 2, 157-167 (1988).

[17] Shepanski, J., and Macy, S., "Teaching Artificial Neural Systems to Drive: Manual Training Techniques for Autonomous Systems," in *IEEE Conference on Neural Information Processing Systems*, D. Z. Anderson (ed.), 693-700 (1988).

[18] Thesen, A., and Lei, L., "An Expert System for Scheduling Robots in a Flexible Electroplating System with Dynamically Changing Workloads," *Proceedings of the Second ORSA/TIMS Conference on FMS: Operations Research Models and Applications*, 555-566 (1986).

[19] Thesen, A., and Lei, L., "An Expert Scheduling System for Material Handling Hoists," *Journal of Manufacturing Systems*, 9:3, 247-252 (1990).

[20] White, H., "Some Asymptotic Results for Learning in Single Hidden-Layer Feedforward Network Models," *Journal of American Statistical Association*, 84:408, 1003-1031 (1989).

[21] Yih, Y., "Trace-Driven Knowledge Acquisition (TDKA) for Rule-Based Real-Time Scheduling Systems," *Journal of Intelligent Manufacturing*, 1:4, 217-230 (1990).

[22] Yih, Y., "Learning Scheduling Rules for FMS from the Optimal Policy of User-Based Semi-Markov Decision Processes," *Proceedings of The Fourth International Conference on Expert Systems in Production and Operations Management*, Hilton Head

Island, South Carolina, 175-183 (May 14-16, 1990).

[23] Yih, Y., and Thesen, A., "Semi-Markov Decision Models for Real-Time Scheduling," *International Journal of Production Research*, 29:11, 2331-2346 (1991).

Appendix A.

Procedure to Avoid the Basic Tank Conflict

```
Temp := 0;
Shift := 0;
For Tank_no := 1 to 4 do
Begin
  Temp := Temp + Previous_ptime[i+1]
    - Current_ptime[i];
  Shift := Max ( Shift, Temp);
End;
```

where

Shift : Time interval needed for avoiding tank conflict
Temp : Temporary variable
Previous_ptime[i] : Minimum processing time for tank i of the previous job
Current_ptime[i] : Minimum processing time for tank i of the current job
Max (a,b) : Function returning the maximal value of a and b.

For example, if job 15 started the process at time t

	Minimum Processing Time in Each Tank				
Job No.	Tank 1	Tank 2	Tank 3	Tank 4	Tank 5
15	23	16	33	25	31
16	18	29	32	22	21

Then, $16 - 18 = -2$ Shift $= 0$
$-2 + (33-29) = 2$ Shift $= 2$
$2 + (25-32) = -5$ Shift $= 2$
$-5 + (31-22) = 4$ Shift $= 4$

The time interval needed is 4, so the earliest time could start job 16 without conflict in usage of chemical tank is $(t + 4 + 23)$.

Yuehwern Yih is an Assistant Professor of Industrial Engineering at Purdue University. She teaches and conducts research in the areas of knowledge-based systems, production control, real-time scheduling, and inductive learning. Her current research projects include the development of machine learning algorithms, and the integration of OR and AI techniques for intelligent scheduling systems. Dr. Yih has published articles on expert scheduling systems, knowledge acquisition, and learning algorithms in journals such as *International Journal of Production Research*, *International Journal of Computer Integrated Manufacturing*, *Journal of Intelligent Manufacturing*, *Journal of Electronics*, etc. She is a member of Omega Rho Honor Society, IIE, AAAI. and TIMS.

Ting-Peng Liang is an Associate Professor at the Krannert Graduate School of Management of Purdue University. He received his Ph.D. degree from The Wharton School of the University of Pennsylvania and MBA degree from the National Sun Yat-Sen University in Taiwan, ROC. Prior to joining Purdue University, he was an Assistant Professor at the University of Illinois at Urbana-Champaign. Dr. Liang has published in journals such as *Management Science*, *Operations Research*, *MIS Quarterly*, *Decision Science*, among others. His major research interests include model management for decision support, machine learning, knowledge acquisition, and the integration of AI, OR, and statistical techniques.

Herbert Moskowitz is the Lewis B. Cullman Distinguished Professor of Manufacturing Management and is Director of the Center for the Management of Manufacturing Enterprises at the Krannert Graduate School of Management at Purdue University. His area of specialization is management science and quantitative methods, quality control, and decision analysis. He is the co-author of four texts, and has published over eighty articles in the areas of decision making, optimization, management science, and quality control in refereed academic journals. He is an associate editor of *Decision Sciences*, the *Journal of Behavioral Decision Making*, the *Journal of Interdisciplinary Modeling and Simulation*, and has been a special associate editor of *Management Science*. He is a Fellow and past Secretary of the Decision Sciences Institute and also served as a Vice-President and Member of the Executive Board of this organization. He also holds positions with TIMS and ORSA, and is a member of the American Society of Quality Control Standards Council.