CHAPTER 9

Learning to schedule and unbalance production using simulation and rule induction

R.M. Kerr and D. Kibira

Department of Industrial Technology and Management,
The University of New South Wales, Sydney, 2052, Australia

Abstract
This paper describes experiments performed to demonstrate the feasibility of applying human learning and machine induction to reactive scheduling of an unbalanced telephone production line in a simulated environment. The simulation model is used to accelerate human training, to log the scheduling decisions of a so trained human expert, and to form the basis for the generation of machine learned decision rules which replicate the human's ability to optimise the performance of the plant.

Keyword Codes: H.1.2, I.2.1
Keywords: User/Machine Systems, Applications and Expert Systems

1. INTRODUCTION

The complexity and variability in manufacturing systems makes it difficult to develop automated scheduling systems by using analytical methods. To manufacture competitively however, it is important to react rapidly to changes in the manufacturing environment, using current shop floor status information so as to minimise the effect of the disruptions (Bengu, 1994). This paper reports an approach to reactive scheduling in a telephone manufacturing plant by using machine learning from a human scheduler already skilled in the task. It is assumed that the human is capable of developing what might be termed "satisficing" schedules which obey the major constraints in the context of a more realistic model of the plant as perceived by the human. A schedule is developed using a set of empirical "rules of thumb" based on observation of the effects of different scheduling decisions over a fairly lengthy "learning" period. As the conditions in the plant are always changing, the learning process is continuous.

A simulated model of a plant is used to aid heuristic search for a better schedule than that obtained by a human scheduler in two ways:

(i) Fine tuning of a human derived schedule using the simulation and adjusting by trial and error to get a better schedule. However, the delay involved in conducting the trial and error search makes it unsuitable for use as a real-time scheduling tool.

R. Kerr et al. (eds.), *Artificial Intelligence in Reactive Scheduling*

(ii) Simulating the plant and hence speeding up the human learning process by using rule induction rather than reality on which to perform the physical observations that generate the rules. A simulation model has been used in preparing a training set for input to an induction machine.

The human expertise captured in the form of an induced set of scheduling rules were subsequently tested by running the simulation in automatic scheduling mode with the schedule being generated entirely by the induced rules. This paper describes the procedure followed, the problems encountered and how they were solved and discusses possible future work.

2. DESCRIPTION OF THE TELEPHONE PRODUCTION LINE

The resources on the shop floor are organised as a flowline consisting of a series of workstations which execute different operations that define the overall manufacturing process. A simplified schematic view of the production process operation is shown in Figure 1. High speed automatic equipment initially insert about 90% of the components into printed circuit boards (PCBs), with the remaining odd shaped components inserted manually. The printed board assemblies (PBAs) are then wave soldered and after solder touch up and manual inspection are tested for defects such as missing components. Defectives are repaired and retested. A keypad is next assembled to the PBA and routed to a case assembling process where the board is assembled to a case moulding. Boards are now routed to a phone testing process. Exterior mouldings are assembled to the case at base assembly and then conveyed to packing where assembled and tested handsets are packed together with phones in the same box. A belt conveyor links the various workstations.

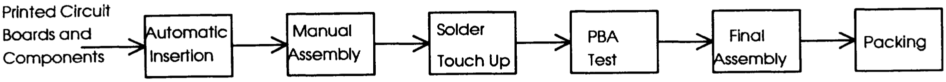


Figure 1. Basic stages in the manufacturing process.

PBAs are the major entities in the simulation model that was developed for this plant. Printed circuit boards arrive on a JIT principle which is modelled by using the current entity to activate the generation of a new entity as soon as it completes processing at an automatic insertion machine. Consequently, the supply to the first machine is assumed to be finite and uninterrupted. However, a counter is maintained so that the creation of new boards in halted at a desired finite number. The number of boards pushed into the system depends on the start-of-shift buffers, desired end-of-shift buffers and volume throughput.

Two products types may simultaneously be processed on the production line. There are three PBA test machines and PBA test machine no 1 is semi-dedicated to product 1 and machine no 2 to product 2 Machine 3 can process either of the two but this resource will be allocated to the product type with more pieces required to complete an order of that type for the shift. When an order of a particular type has completed processing by the bottleneck resource, the machine dedicated to it is allocated to process the other product. If the orders for both product types have been fully processed at PBA test these resources become idle. At other workstations, work in progress of different products is processed approximately in the proportion of part type 'demand' downstream which depends on the volume throughput required for the shift, current buffer levels and required target buffer levels. Each product has different process times at some workstations.

Although boards and inserted components arrive at the different stages in the production process by using a JIT principle, work-in-progress inventories are not entirely eliminated for such reasons as: work orders may arrive randomly at the plant, some operators can be absent from work without notice, machine breakdowns are random, processing times are not fully predictable. This type of production environment necessitates some level of safety stock at some or all workstations, however JIT allows for an effective control of this buffering to eliminate excessive build up.

The slowest workstation on the line is the PBA test and the task of scheduling is to allocate labour resources to workstations upstream of this workstation and at handset assembly and test workstation to avoid build-up of buffer stocks and maximise volume throughput in response to process time variability, component failure, machine breakdowns, planned maintenance, and labour absenteeism. Though throughput essentially depends on failure characteristics of machines and especially PBA test, it is shown through a series of simulation experiments that the maximum output volume expected from the line is approximately 2600 pieces. The scheduling task is essentially that of dynamic line balancing by determination of an appropriate labour allocation on the line. The objective of the scheduler is to improve volume throughput while reducing inventory levels under circumstances of variability due to machine breakdowns, component failures and labour absenteeism. Variability here results in large fluctuations in buffer levels, volume throughput and labour requirements. Thus the objectives of the scheduling task are:

(a) To minimise and stabilise labour requirements.
(b) To minimise total work in progress.
(c) To maximise and stabilise volume throughput.
(d) To improve delivery performance.

3. DETERMINISTIC SCHEDULING APPROACH

This research assumes that human " expertise" is an essential component of the scheduling task, and that this expertise can be learnt by intensive training using a simulation model. However, if the production line is treated as a deterministic system, the labour allocation task can be handled analytically using a set of equations which define the evolution of buffer stocks as a function of time and of the processing rates at each workstation. It was assumed that the human's ability to make reactive decisions on capacity reallocation in the face of disturbances using a qualitative but much "richer" model of the behaviour of the system than that underlying the deterministic approach is the main reason why a human can develop better schedules. In order to test this basic assumption and provide a reference point for developing human expertise, a simple deterministic scheduler was developed with the purpose of comparing the schedules so produced with those prepared by an "expert" human.

The deterministic capacity allocator treats the telephone line as a series of sequential workstations with the production rates depending on the number of operators allocated, separated by buffer stocks. Buffer sizes are assumed to be a linear function of time and of the production rates of the workstation immediately upstream and downstream of the buffer.

If workstation i-1 feeds workstation i at a rate r_{i-1}. The buffer size at workstation i at time t, denoted by $b_i(t)$ is related to the size of buffer at time 0, $b_i(0)$ by the equation:

$$b_i(t) = b_i(0) + (r_{i-1} - r_i)t \qquad (1)$$

or $$b_{i+1}(t) = b_{i+1}(0) + (r_i - r_{i+1})t$$

thus $$r_i = r_{i+1} + (b_{i+1}(t) - b_{i+1}(0)) / t \qquad (2)$$

This expression is evaluated for each product type to get the total processing rate required at each workstation. Scheduling software based on these concepts (termed the "capacity allocator") was written in the C programming language. Figure 2 shows a plot of the total volume throughput when the start-of-shift state at the beginning of each simulation run is different and selected by fractional factorial design of experiments (Franklin, 1984; Montgomery, 1991) to test the performance of the deterministic capacity allocation program. Final buffers were targeted to a minimum levels and the measure of performance used is the total volume throughput. Improper placement of resources upstream of the bottleneck resource on the production line can result in work starvation and hence a lower utilisation of resources and a reduced volume throughput for a frugal resource placement on the one hand or excessively high work-in-progress buffer stocks at some workstations for a lavish usage of available resources, on the other.

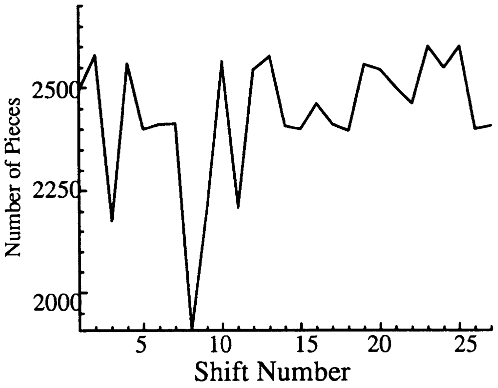


Figure 2. Total volume throughput at the end of each simulated shift.

Figure 2 shows that the total volume throughput for each simulated shift is well below the expected volume of about 2600 pieces and to improve line performance for this measure of performance an attempt is made to use human learned heuristics as will be described in the next section.

4. LEARNING SCHEME

The training to achieve human expertise was carried out by using the simulation model, which for many complex tasks such as flying a plane, etc. is the first phase in training (Michie, 1990). A model of the production line, as described, was written in the SIMAN simulation language with graphics animation in CINEMA. Due to the complexity of the operations and variability in the simulated line it is considered here as a 'black box' (Sammut *et al.*, 1991) where the effects of labour allocation on buffer levels and throughput must be inferred by experimentation rather than by classical formulation of the scheduling problem as is illustrated in Figure 3. The procedure in the figure is repeated on several sets of starting and target buffer levels until a satisfactory and consistent skill is attained. Capacity allocation on the line is such that buffer levels are kept to a minimum, however, the rules developed by the human are robust enough to schedule the line such that a buffer at any workstation can be targeted to any level within the operating domain cover such events as a planned shutdown of the upstream workstation for maintenance.

When a human needs to make a scheduling decision during a shift operation, he hits the escape key and the simulation run is automatically stopped. By picking the 'Debug aids' from the menu by using a mouse, the number of operators at any workstation can be changed during the shift. The user can advance the simulation clock in time with the simulation running in background mode (if desired) or to clear the systems to restart the simulation run or use any other SIMAN/CINEMA run-time facilities.

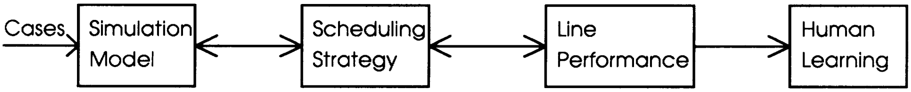


Figure 3. Human Learning Scheme.

4.1 Human Derivation of Schedules

Scheduling decisions are carried out using linguistic terms to describe the magnitudes of buffer sizes: *very small, small, medium, large* and *very large*. The human scheduler uses visual observation and experience to classify the number of pieces into one of the above categories based on the length of a bar proportional to the buffer size recorded by the simulation and displayed by the CINEMA graphics. The guiding principle in learning how to allocate labour is the continual adjustment of production rates in such a way as to restore interprocess buffers to their target values by the end of the shift. Starting with the terminal workstation on the line and working backwards, a suitable allocation is determined considering the total labour capacity available. The same principle is applied at any labour reallocation point in time during the shift. An initial allocation is determined at the beginning of the shift with reallocation of labour if there are any breakdowns of the bottleneck resource or if the buffer level at any workstation exceeds or falls far below the target level. The frequency of labour changeovers is limited by constraining a particular configuration of labour to be in force for at least 60 minutes.

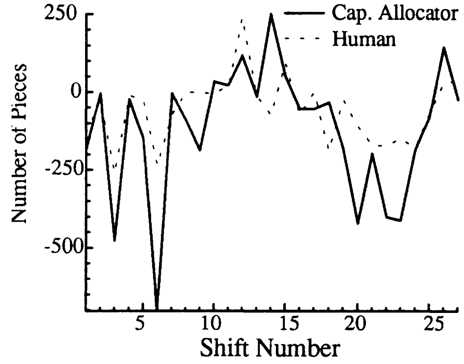
The general scheduling strategy can be summarised as follows:

if beginning of a shift or
if end of scheduled breaks or
if any buffer level exceeds a given threshold level compared with the target at any time or
if any buffer level falls below a given threshold level compared with the target at any time or
if there is any equipment breakdown or
if the equipment previously broken down has finished repair or
if there is sudden drop in the total labour capacity availability on the line then
reallocate labour on the entire production line on the basis of current buffer levels, additional phones to be produced, time into the shift, equipment status and total labour capacity availability.

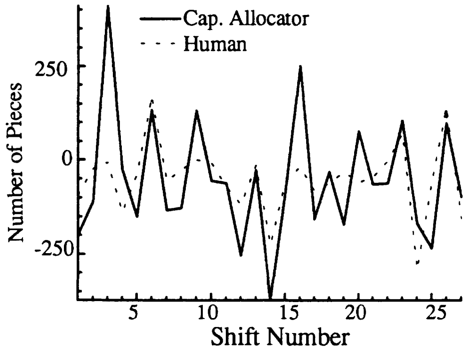
From this logical rule structure it is seen that the learning system does not use any deep knowledge of the manufacturing system that it was scheduling or of its underlying simulation code other than its response to a set of labour placement at the different workstations.

4.2 Experimental Set Up and Comparison of Capacity Allocator and Human derived Schedules

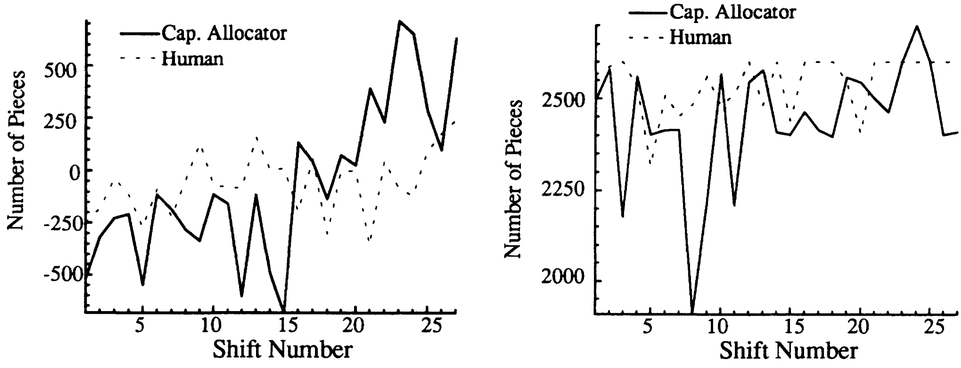
Human performance was compared to capacity allocator schedules using a series of experiments. The buffers which determine the operating characteristics of the production line are those at the workstations upstream of the bottleneck resource but for these experiments a three-level fractional factorial experiment was developed to investigate buffers at PBA test machines, of handsets at packing and volume throughput of each product in order to compare the performance of the capacity allocator and rules derived by human learning. Figure 4 shows a graphical comparison of the differences in final buffers between target and the actual final levels for capacity allocator with human learned heuristics.



Differences in buffers at PBA test for type 1



Differences in buffers at PBA test for type 2



Differences in buffers of handsets at packing

Volume throughput

Figure 4. Comparison capacity allocator and human derived schedules.

The closer a graph is to the zero line, the better is the schedule performance. Figure 4 shows that capacity allocator is poor at scheduling the line in presence of the variability and complexity associated with a real world manufacturing system. A human, on the other hand, after some learning has developed better schedules against the same measure of performance. We, thus, conclude that a valid technique for automatic scheduling of this line is for machine to induce these human learned heuristics.

5. DEVELOPING THE AUTOMATED RULE BASED SCHEDULER

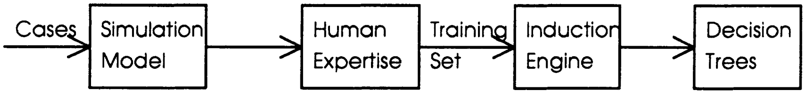


Figure 5 Machine Induction Scheme.

The essence of induction is to move from a known set of examples to a set of general rules that summarise the information in the examples supplied and hopefully, other cases not included in the training set but can be represented by the same attributes.

Figure 5 schematically illustrates the machine induction scheme used. The inductive inference tool used for these experiments is C4.5 (Quinlan, 1993), a descendant of ID3 (Quinlan, 1979) on a training set supplied by a human. Rule induction is one way around the so called 'knowledge acquisition bottleneck' that is encountered during the development of knowledge based systems (Michie, 1987). It often arises because experts have difficulty in articulating their expert knowledge as a set of rules.

The induced set of rules forms a basis for the automatic scheduler, which should be run as follows:

(a) At the start of every shift to determine an initial allocation of labour for that shift, on the basis of current buffer levels.
(b) On machine breakdowns.
(c) If any buffer level exceeds or falls far below the required target level.
(d) When there is a sudden drop in total labour capacity available.

The following inputs are used:

(a) Buffer stocks at the beginning of the shift.
(b) Required buffers at the end of the shift.
(c) Status of equipment resources.
(d) The total labour capacity available or anticipated to be available at any time during the shift.

The task of the automatic scheduler is to determine an appropriate labour placement at each workstation, subject to the above conditions and constraints so as to meet the production target of each telephone type and to maintain the target buffer sizes during a shift.

5.1 Logging Scheduling Information

The simulation model was written using the different SIMAN and the C programming routines such that when a human relocates labour on the line during a shift, the prior state represented by buffer levels at that time and time during a shift are written to file. The training set was obtained from 200 cases. Four decision trees are constructed using the C4.5 induction program, one for each of the workstations: Automatic Insertion, Manual Assembly, Solder Touch Up and handset assembly/test. For each individual workstation the attributes are as shown in Figure 6. The dependent variable or class value in each case is the labour allocation for that workstation. A program has been written to directly convert C4.5's decision trees into if-statements in C so that they can easily be incorporated into the simulation model.

Workstation	*Relevant Attributes*
Automatic Insertion	(a) Differences in buffer levels at: - Manual Assembly - Solder Touch Up - PBA Test (b) Number of phones left to meet shift production target (c) Time into the shift at the reallocation point
Manual Assembly	(a) Differences in buffer levels at: -Solder Touch Up -PBA Test (b) Number of phones left to meet shift production target (c) Time into shift at the reallocation point
Solder Touch Up	(a) Differences in buffer levels at: -PBA Test (b) Number of phones left to meet shift production target (c) Time into shift at the reallocation point
Handset Assembly and Test	(a) Differences in buffer levels of: -Handsets at packing (b) Target throughput

Figure 6. Attributes relevant to labour allocation.

5.2 Using the Automatic Scheduler

After the decision trees to schedule and control the simulation of the production line have been synthesised they are tested by replacing the expert the human scheduler's task with the decision tree code.

Start-of-shift and required end-of-shift levels and volume throughput are entered via file and read into the simulation model, and the automatic scheduler then determines an allocation of labour on basis of these buffer levels. In case of contingencies or end of scheduled breaks during this shift, a new set of labour allocation corresponding to the conditions prevailing is determined. The shift is divided into three stages since the buffer sizes expected are of different sizes as time advances and the by using if statements to check the time, the automated scheduler switches to a different set of rules.

Results show that the automatic scheduler allocates labour capacity in a manner very similar to that of the human scheduler who provided the data from which the rules were induced. It is demonstrated here how the rules operate by describing one decision tree.

The first few branches of a decision tree for the allocation of labour at the automatic insertion workstation in the first stage:

```
if no of pieces of type 2 = Very Small then
   if no of pieces of type 1 = Very Small then allocation = 0
else if no of pieces of type 2 = Very Small then
   if no of pieces of type 1 = Small then allocation = 1
else if no of pieces of type 2 = Very Small then
   if no of pieces of type 1 = (Medium or Large) then allocation = 2
else if no of pieces of type 2 = Very Small then
   if no of pieces of type 1 = Very Large then allocation = 3
-
-
-
```

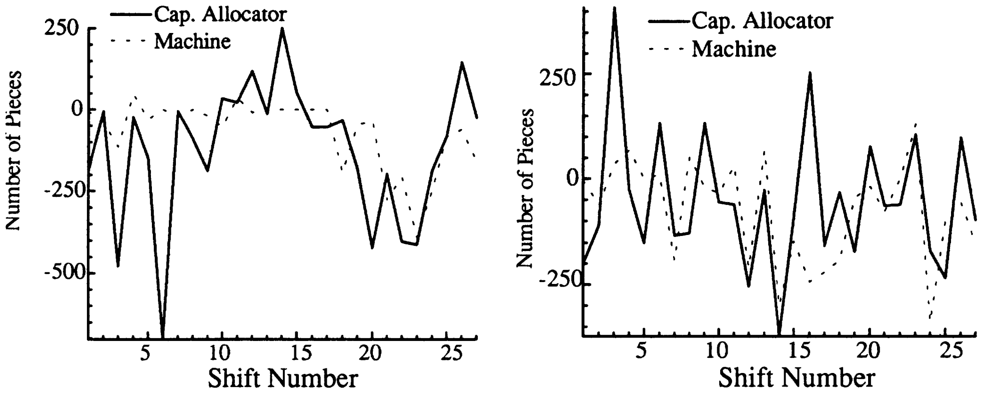
These expressions define linguistic relationships between a set of no of pieces of type 1 and pieces of type 2 and labour placement for that particular system state. These were mapped into numerical relationships for reading by the simulation as in the following example:

For this workstation at the beginning of shift:

No of Pieces < 100:	*Very Small*
No of Pieces >= 100 & No of Pieces < 400:	*Small*
No of Pieces >= 400 & No of Pieces < 800:	*Medium*
No of Pieces >= 800 & No of Pieces < 1200:	*Large*
No of Pieces >= 1200:	*Very Large*

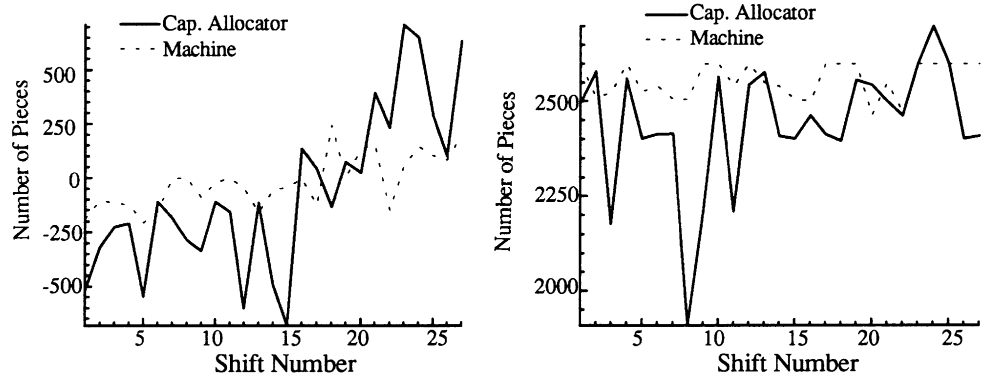
6. COMPARISON OF CAPACITY ALLOCATOR AND MACHINE DERIVED SCHEDULES

In Figure 7, capacity allocator schedules are compared with machine derived schedules for buffers at PBA test machines, handsets at packing and total volume throughput. Fractional factorial design experiments were again used as in the previous sections.



Differences in buffers at PBA test for type 1

Differences in buffers at PBA test for type 2



Differences in buffers of Handsets at packing

Volume throughput

Figure 7. Comparison of capacity allocator and machine derived schedules.

From the graphs above, it can be seen that for all buffers and the total volume throughput, human derived rules obviously perform better than capacity allocator and at least as good as the human who supplied the training set from which these rules were induced.

7. DISCUSSIONS AND CONCLUSIONS

These experiments show that it is feasible to learn to schedule a dynamic manufacturing system where records of state of the system are entered into an induction program which constructs rules that not only automatically determine suitable labour allocation on the basis of previous data but also perform better than analytical formulation which assumes an ideal, deterministic environment. The data provided to the induction program are logs of the actions taken by an experienced human in response to some changes in the system. An induced rule-set constitutes a "strategy" for the given sub-task - a kind of classifier that maps state records of the manufacturing system into labour allocations where causal relations are not considered.

The decision trees so induced can 'satisfactorily' control the operation of the plant as long as its operation keeps within the bounds which were characteristic of it during rule induction experiments. In case one or more of its operating parameters significantly deviate from the original ones, the system then needs to be simulated again and through rule induction, a new set of rules generated. The factors that can lead to these changes can be either internal or external to the manufacturing system. The methodology demonstrated in this paper, however, has the advantage that in case of anticipated changes, simulation experiments and learning be undertaken by the system prior to the changes actually taking place in reality. Current research is aimed at producing a method of building an automatic scheduling code by machine learning of the specified task, although a general strategy would be the building of a generalised method that can be used in a variety of tasks.

This approach provides an innovative complement to the traditional sources of knowledge used in the development of expert systems and should prove useful in the building of real-time control strategies in any manufacturing environment. The application of this methodology can potentially be generalised to more comprehensive scheduling problems in manufacturing. This method is developed and tested in a simulated environment and it would be of interest how its performance would be if the decision rules so obtained were to be tested in the real-life plant. Obviously, the approach of this paper is limited to those cases where it is possible to capture the full complexity of the plant in a simulation model.

REFERENCES

G. Bengu, Simulation-based scheduler for flexible flowlines, *International Journal of Production Research*, (1994), **32**(2), 321-344.

M. F. Franklin, Constructing tables of minimum aberration p^{n-m} designs, *Techtronics*, (1984), **26**, 225-232.

D. C. Montgomery, *Design and analysis of experiments*. John Wiley & Sons, New York, (1991).

D. Michie, M. Bain and Hayes-Michie, Cognitive models from subcognitive skills, *Rule Based Systems in Industrial Control*, Peter Peregrinus Ltd., (1990), 71 - 99.

C. Sammut and D. Michie, Controlling a black box simulation of a spacecraft. *AI Magazine*, (1991),

12(1), 56-63.

J. R. Quinlan, *C4.5 Programs for machine learning,* Morgan and Kaufmann Publishers, Inc., (1993).

J. R. Quinlan, Discovering rules by induction from a large collection of examples, *Expert Systems in the Micro Electronic Age*, (D. Michie, Ed.), Edinburgh University Press, (1979).

D. Michie, Current developments in expert systems: *Proceedings of the second Australian conference on applications of expert systems*, Sydney, (1987), 137-155.