Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

SIMULATION-BASED CAPACITY PLANNING AND SCHEDULING WITH AUTOSCHED

John Everton

AutoSimulations
655 Medical Drive
Bountiful, Utah 84011, U.S.A.

ABSTRACT

The function of 'scheduling' is the time-sequenced allocation of resources, such as machines, personnel, tools, and component materials, to perform a collection of tasks to manufacture products.

Traditional methods of scheduling have used infinite-capacity and static-time calculation approaches. The result of these methods are inaccurate and non-representative answers to extremely important questions. Discrete event simulation can provide a very accurate and effective engine to address scheduling issues. This paper introduces *AutoSched*, a discrete-event, simulation-based, finite-capacity model, as a tool for scheduling equipment and personnel. A sample manufacturing system will be used to illustrate both the issues to be considered and techniques used in scheduling a factory.

1 AUTOSCHED'S VIEW OF THE FACTORY

AutoSched sees the factory as a number of **stations**. Stations can be machines, work benches, assembly positions, or any location where work is performed on a product. A group of stations that perform essentially interchangeable work is called a **family**. Every station belongs to a family. Families share a common input queue and work list for parts waiting for service from one of the stations in the family. Each station can have one or more calendars associated with it. **Calendars** specify when stations are unavailable for work.

Lots flow between families in *AutoSched.* Lots consist of a quantity of pieces of a given type of part, and flow according to a **routing** that you define. A routing consists of a number of steps that represent the process required to produce the part. Each step defines the manufacturing requirements, such as the station family, the setup, the processing time, and the operator class.

AutoSched is extremely flexible in that you only have to provide data that is available and important to you. For example, if you do not want to include the detail of modeling human operators, don't include it. In addition, *AutoSched* has defaults for almost all of the possible features.

Lots can start the simulation either at the first step of their routing, or at their current step if they are in process. As lots flow through the steps in the routing, they enter the family work list and queue for the family designated for that step. Idle, available stations in the family wake up one at a time and execute task selection rules. **Task selection rules** are the criteria each station uses to determine which lot to work on next. The task selection rule allows the station to either pick a lot from the potential parts or to wait for a better choice. The status of operators, tools, components, and other constraints may be considered in a task selection rule. Task selection can help a scheduler to improve factory performance.

1.1 Data Requirements

The data required for *AutoSched* can be entered from exiting databases or spreadsheets, or it can be entered through *AutoSched's* powerful edit tables. An edit table enables you to input or modify data in a stand-alone or integrated database fashion.

1. **Data Organization** - You simply organize the data to define the model; you don't program.
2. **Decision Orientation** - Stations and operators, rather than orders, make the decision of what to work on next. This reflects how decisions are made in the real world, i.e., by operators who can look at the whole system.
3. **Calendar Capability** - The simulation clock is converted to a calendar clock (Month, Day, Year, Hour, Minute, and Second). You can define an unlimited number of calendars and attach them to equipment and personnel. Calendars include information such as scheduled maintenance and holidays.
4. **Schedule Diagnostics** - *AutoSched* provides Gantt charts and business graphs to help you visually interpret the schedules. Because *AutoSched* is based on *AutoMod*, AutoSimulations' 3-D graphical simulation software, animation also aids in the understanding of scheduling dynamics.

5. **Customization** - You can either use *AutoSched's* built-in scheduling rules or customize them using *AutoMod's* flexible simulation language.
6. **Flexibility** - With *AutoMod* and *AutoSched*, the same tool can be used for:
   - Factory simulation
   - Pre-planned schedule creation
   - Real time dispatching
   - Finite capacity planning and analysis

A scheduling or planning model requires data such as: routing definitions (for each part type), stations, operators, tools, orders and WIP status, and calendars. Much of this information may exist in a database such as an MRP or a related CIM package. If it does exist, you don't need to duplicate the information, as it can be imported into *AutoSched*. The following is a brief explanation of *AutoSched's* files:

Routings - The routing information describes the process steps of the different products. Each product type may have its own routing or may share a routing with another product.

Stations - Each station is defined in this file. A station is defined as a place where work is performed. Stations are grouped into families. A family is a group of stations that share a common input queue.

Operators - Each operator is defined in this file, along with skill categories and certification levels.

Tools - This information describes each of the factory's tools/fixtures.

Order and WIP Status -The order status file describes all orders scheduled during the simulation period and their current status. This information consists of both a current 'snapshot' of the shop floor and a list of the orders yet to be released.

Calendars - Each resource (stations, operators, tools) can have its own work schedule. This work schedule can be composed of an unlimited number of calendars. This file describes the different types of calendars. Once they are defined, they can be attached to resources in the attachment file. There are five calendar types: shift, down, preventative maintenance (PM), holiday/exception, and reserved.

Calendar Attachment - A resource can have any number of calendars attached to it. The order in which calendars are attached to the resource dictates the order of precedence. For example, an operator might have a shift calendar, a holiday calendar, and a vacation calendar attached to it respectively. The holiday calendar overrides the common periods with the shift calendar, and the vacation calendar overrides the common periods with the other calendars. Calendar attachments allow a flexible, easy-to-maintain work schedule.

1.2 Scheduling Rules

AutoSched uses a revolutionary approach to scheduling rules. This approach makes it possible for plant scheduling personnel to define rules for each resource, such as stations and human operators. These rules are easy to construct, and are not limited to a single criteria or a single view of a resource.

As shown in Figure 1, a rule is a series of filters through which potential tasks are passed. Each filter is a criterion, or test, the lots must pass. The filters screen out more and more lots until either one or no lots (or batches) remain for final selection.

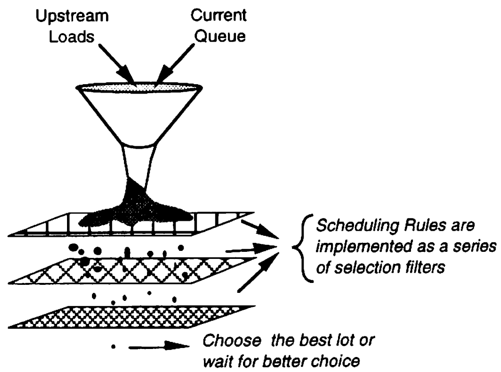

Figure 1: Decision Filter

You can utilize decision tree, sorting, and filtering capabilities in these rules. A rule can contain as many filters as needed.

This approach allows rules to consider multiple criteria in the decision process, thus allowing better decisions to be made.

1.3 What Does a Scheduling Rule Look Like and How Does it Work?

Scheduling rules consist of one or more logic filters. Potential lots must pass through these logic filters to be selected. Figure 2 shows an example of a scheduling task rule. This example is simple. However, there is no limit to the level of sophistication that can be used in a scheduling task rule.

Example:

Suppose you must develop a scheduling task rule for a station where the amount of setup changeovers must be minimized for quality reasons or due to a lengthy setup time. When it's necessary to change to a new setup, the station should not select a lot or a batch of lots that requires a setup that is being used by any other station within the same family. If there is more than one lot or batch that

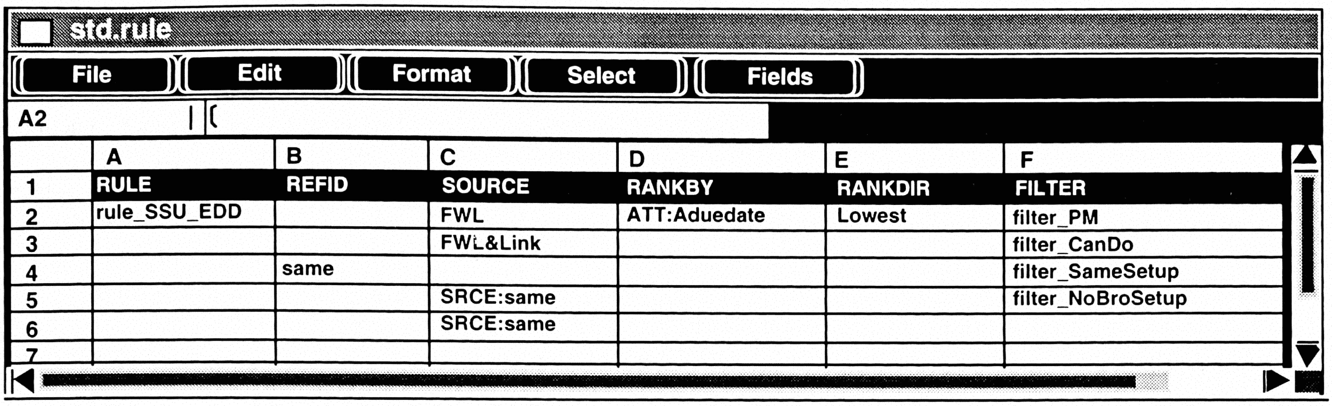

Figure 2: Scheduling Task Rule

meets the criteria, the one with the earliest due date is chosen. The decision tree demonstrates the filtering logic. The station executes this logic when it finishes a task or after incoming work awakens it.

These rules do not guarantee the optimum schedule; no one can guarantee the optimum schedule due to the large number of possible combinations. However, through experimentation, rules can be developed that improve the performance of the factory on the order of 35% - 60% (Norman, 1989).

Before developing scheduling rules, however, the factory's management must ask an important question: What is a good schedule?

1.4 What is a Good Schedule?

While this appears easy to answer, it's often more difficult to quantify than you might think. The attributes of a good schedule are:

- Minimum lead time
- On-time completion of all orders
- Minimum work-in-process and finished goods inventory
- Maximum resource utilization
- Minimum or no overtime
- Minimum cost in terms of the routing through alternate machines

While these are all worthy goals of a schedule, the fact is that some of them conflict. For example, assume your company just purchased a new, flexible machining center that cost $250,000. To maximize the investment, the plant manager mandates that the new machining center must be utilized more than 90% of the available time.

To utilize equipment to that extent, there must constantly be a queue of work in front of it, and when one order is completed another one must be waiting. Actually, several orders must be waiting, in case of unforeseen situations. In this example, the utilization goal conflicts with lead time, inventory, and possibly on-time completion goals, because products that must be serviced by the new machining center wait in a large queue.

Therefore, the answer to the question, "What is a good schedule?" is: conformance to management's goals and objectives. These goals and objectives must be developed by understanding inherent conflicts and the dynamics of relationships in a system. They can be measured by:

- Due date performance (the measure of earliness or lateness)
- Throughput (number of products completed per unit time)
- Lead time (amount of time orders stay in the system)
- Inventory levels (WIP and Finished Goods)
- Resource utilization (percent of time a resource produces)

AutoSched easily computes these and other performance measures for each schedule produced.

2 DESCRIPTION OF SAMPLE MANUFACTURING SYSTEM

Our sample manufacturing system has the following components. *AutoSched* uses keywords at the top of each column to define the data contained in the column.

Stations (equipment)

FAMILY	QTY	BATCHMIN	BATCHMAX
saw	6	1	1
die attach	12	1	1
wire bond	25	1	1
die coat	4	1	1
encapsulation	8	1	1
laser	2	1	1
deflash	2	1	1
trim & form	4	1	1
lead finish	8	1	1

The cure process uses an oven and can process from 1 to 10 lots at a time. Parts are grouped by part type, and the processing time is the same regardless of how many lots are being processed.

Parts

There are six different parts. Each use the same routing or process and processing times. Each have a default lead time of 1 day.

Routing (process)

STEP	FAMILY	PTIME	PTUNITS	PTPER	STIME	STUNITS
100	saw	1.0	sec	piece	10	min
200	die attach	0.8	sec	piece	20	min
300	cure	2.0	hr	lot	1	hr
400	wire bond	3.6	sec	piece	20	min
500	die coat	2.0	sec	piece	10	min
600	cure	3.0	hr	lot	1	hr
700	encapsulation	1.4	sec	piece	18	min
800	laser	0.7	sec	piece	10	min
900	deflash	0.6	sec	piece	10	min
1000	cure	6.0	hr	lot	1	hr
1100	trim & form	1.0	sec	piece	5	min
1200	lead finish	0.9	sec	piece	10	min

(PTIME = processing time, PTUNITS = processing time units, PTPER = processing time based on the number of lots or pieces, STIME = setup time, STUNITS = setup time units). Setups times are taken when a machine switches to a new part. The oven requires a setup time based on the part and step; therefore a setup is required for part_a at step 300 to part_a at step 600. The oven load time is part of the processing time.

Calendars

All stations are operated 5 days a week, two shifts per day. There is a 15 minute period between shifts when the machines, except for ovens, do not operate.

Demand

LOT	PART	REPEAT	RUNITS	PIECES	CURSTEP	PRIOR
lot1_1	part1	2	hr	2500	begin	10
lot1_p1	part1	8	hr	2500	begin	1
lot1_2	part2	2.2	hr	2500	begin	10
lot1_p2	part2	8	hr	2500	begin	1
lot1_3	part3	2.4	hr	2500	begin	10
lot1_p3	part3	8	hr	2500	begin	1
lot1_4	part4	1.5	hr	2500	begin	10
lot1_p4	part4	8	hr	2500	begin	1
lot1_5	part5	1.6	hr	2500	begin	10
lot1_p5	part5	8	hr	2500	begin	1
lot1_6	part6	1	hr	2500	begin	10
lot1_p6	part6	8	hr	2500	begin	1

(REPEAT causes a lot to be generated each unit of time. CURSTEP = the current manufacturing step; new lots use begin. PRIOR = priority)

WIP Lots

LOT	PART	REPEAT	RUNITS	PIECES	CURSTEP	PRIOR
lot1_w1	part1	1	secs	2500	400	10
lot1_w2	part2	1	secs	2500	500	10
lot1_w3	part3	1	secs	2500	600	10
lot1_w4	part4	1	secs	2500	700	10
lot1_w5	part5	1	secs	2500	1000	10
lot1_w6	part6	1	secs	2500	1100	10
lot1_w7	part1	1	secs	2500	400	10
lot1_w8	part2	1	secs	2500	500	10
lot1_w9	part3	1	secs	2500	600	10
lot1_w10	part4	1	secs	2500	700	10
lot1_w11	part5	1	secs	2500	1000	10
lot1_w12	part6	1	secs	2500	1100	10

Ten of each wip lot is started. All wip lots are priority 10.

3 SCHEDULING MODEL RESULTS

The scheduling model is used to produce a dispatch or sequence list of lots to work on for each machine. Because the scheduling model uses simulation, it can also used to analyze performance of the manufacturing system. By using different task selection rules, the scheduler can attempt to improve performance. We will focus on the performance measures and compare results using several standard and advanced dispatch rules.

The model runs for 5 days. All processing, setup and lot inter-arrival times are constant.

3.1. Base Model — FIFO

The base model uses the FIFO (first-in-first-out) rule for all stations. The following chart indicates observations of the average cycle time and cycle time of the last observed lot. Observations were made every hour after one day and 15 hours which was when the first non-WIP lot was completed. Statistics do not include WIP lots.

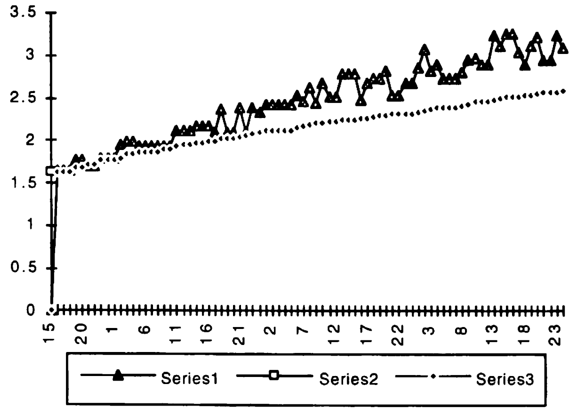

Figure 3: Cycle Time Chart

The cycle time chart indicates that more lots are being started that the system can handle. A lead time of 1 day was desired. The final observation indicates the following cycle time data.

Non-priority lots:

Average cycle time	2.58	days
Lost completed	165	
Max cycle time	3.32	days

Priority lots:

Average cycle time	2.53	days
Lost completed	36	
Max cycle time	3.27	days

The main bottlenecks are:

	% of time processing	% of time setting up	% of time off shift	% of time idle
die coat	90.42	7.5	2.08	0.00
cure	81.95	16.61	0.00	1.44

3.2. First Experiment — Same Setup

The same setup rule can improve performance by reducing the amount of setup time. The rule uses three logical steps:

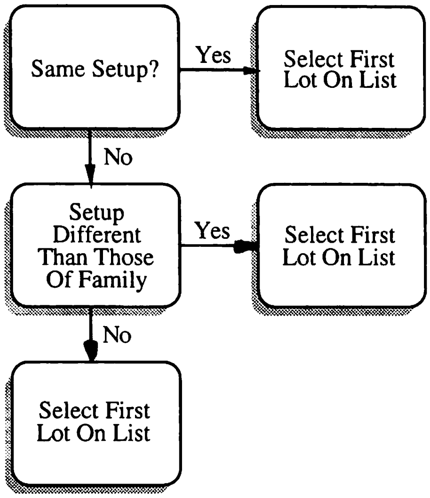

Figure 4: Same Setup

The first filter sorts out all lots that match the current setup. If there are any, the first one is selected. Because no ranking functions are being used, lots will be moved from one list to another using FIFO (the arrival time at the Family Work List). The second filter is unique to *AutoSched*: if a new setup is needed, this filter tries to find a setup that is not currently being run on other members of the family. This will avoid having more than one station running the same setup. Finally, the filter will select the first lot on the list. Another variation of this rule is to look upstream for pending lots before switching. Also, the rule can cause the station to wait for a period of time to see if a matching setup arrives before changing setups.

The model shows improvement using same setup for all stations.

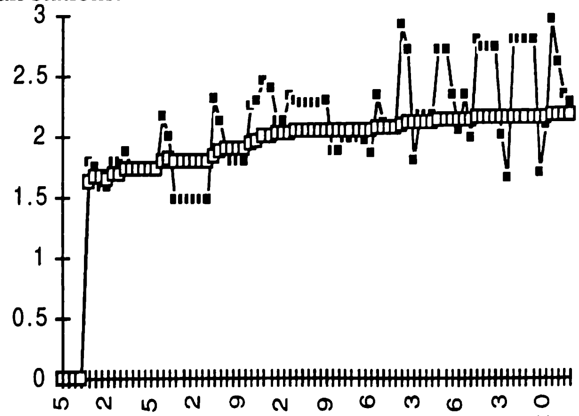

Figure 5: Cycle Time Chart

The cycle time chart indicates a reduction in cycle time. The following results indicate a reduced cycle time, increased processing time, and increased lots completed. Die coat is still a bottleneck (zero idle time) and the cycle time is still growing.

Non-priority lots:	FIFO	SSU	
Average cycle time	2.58	2.16	days
Lots completed	165	192	
Max cycle time	3.32	2.98	days
Priority lots:			
Average cycle time	2.53	2.15	days
Lots completed	36	40	
Max cycle time	3.27	2.88	days

Bottleneck performance improved:

	% of time processing	% of time setting up	% of time off shift	% of time idle
FIFO				
die coat	90.42	7.5	2.08	0.00
cure	81.95	16.61	0.00	1.44
Same Setup				
die coat	97.36	0.56	2.08	0.00
cure	88.26	10.30	0.00	1.44

3.3 Second Experiment — Priority Rule

The priority rule pre-sorts the list and ranks by priority, then selects the first lot on the list. If there is more than one priority lot, the first will be selected FIFO. The manufacturing system uses a 1 for high priority and 10 for normal priority.

The following results indicate a reduced cycle time and increased number of priority lots completed. The total production is less than SSU and about the same as FIFO. Die coat is still a bottleneck (zero idle time) and the cycle time is still growing.

Non priority lots:	FIFO	SSU	PRIOR	
Average cycle time	2.58	2.16	2.69	days
Lots completed	165	192	136	
Max cycle time	3.32	2.98	3.65	days
Priority lots:				
Average cycle time	2.53	2.15	1.6	days
Lots completed	36	40	63	
Max cycle time	3.27	2.88	1.98	days
Total lots produced	201	232	199	

Bottleneck was similar to FIFO:

	% of time processing	% of time setting up	% of time off shift	% of time idle
FIFO				
die coat	90.42	7.50	2.08	0.00
cure	81.95	16.61	0.00	1.44

3.5 Cycle Time Rule — Constraint Based Management

All the experiments so far have indicated a bottleneck in die coat and cure. Let us apply a rule that uses theory of constraints as defined in *The Race* by Goldratt. The first experiment will be to limit the number of lots enroute to or being processed by die coat. The rule for the first process, saw, will not allow lots to start unless the buffer count is below a certain level.

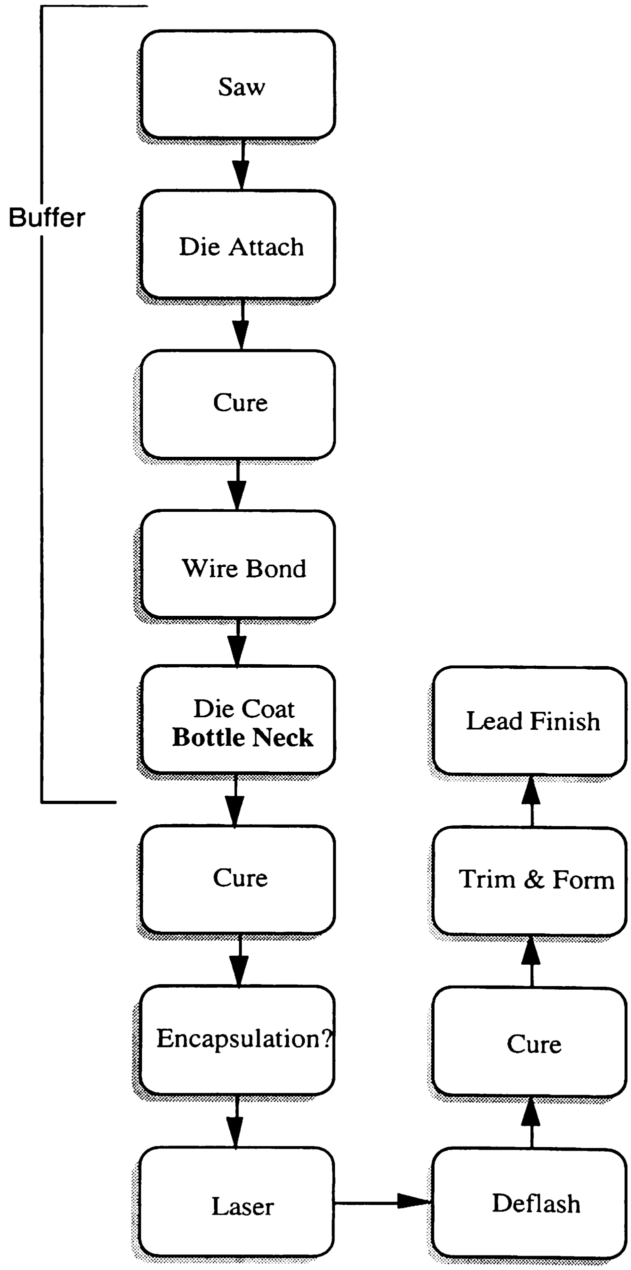

Figure 8: Die Coat Buffer

We will run several iterations of the model varying the buffer limit to determine the optimum limit. Same Setup will be used for all processes since it produced the highest throughput. Priority lots will be ignored since we are trying to develop a smooth flowing system with predictable cycle times. The following table indicates the results by starting the buffer at 100 and increasing it by 10 to 150:

	100	110	120	130	140	150
Lots Completed	126	176	211	220	223	210
Cycle Time (days)	1.54	1.57	1.68	1.78	1.91	2.04
Die Coat -						
Utilization (%idle)	35.12	11.35	2.05	0	0	0

Throughput peaks at a buffer limit of 140 lots and starts to drop after 150. This is because the cure operation is required at three steps. When cure is overloaded, the queue becomes larger which increases the average cycle time and reduces throughput. Even though die coat is the gateing process and is upstream from cure, cure is at a high utilization (1.4 % idle with 150). It is also interesting that throughput continues to increase a small amount when Die Coat utilization reaches 100 percent. This is due to slightly more efficient setups particularly at cure which is used at one of the steps prior to the bottleneck.

Cycle time continues to increase more sharply after a buffer of 120:

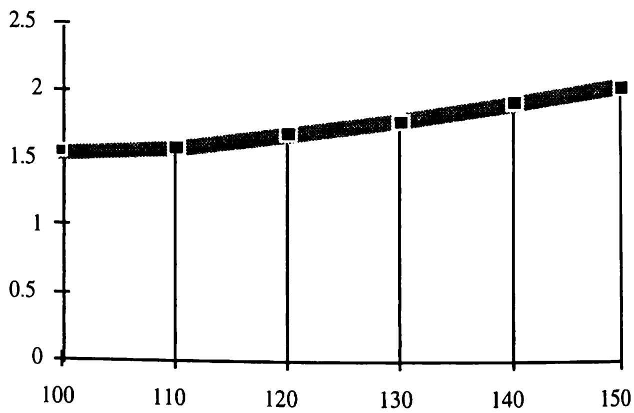

Figure 9: Cycle Time Chart

The theoretical cycle time is determined by adding the processing times for each step and does not include setup time, queuing time, or off shift. In this case, the theoretical cycle time is 19.33 hours. Even with die coat at 35 percent idle, the average cycle time is about 1.5 days. This would suggest the natural cycle time, including setup time, queuing time, and off shift, might be close to 1.5 days.

Same Setup				
die coat	97.36	0.56	2.08	0.00
cure	88.26	10.30	0.00	1.44
Priority				
die coat	89.93	7.99	2.08	0.00
cure	82.85	15.77	0.00	1.38

3.4 Combination Rule — Same Setup & Priority

The power of *AutoSched's* task selection rules is the ability to combine multiple filters and ranking functions into combination rules. Within *AutoSched's* rule editor, you can branch to a different rule or filter. For this experiment, the following combination of Same Setup and Priority will be used. This rule will be referred to as SSU_PR.

Another variation of this combination is to give preference to priority lots. Figure 6 shows the flow chart will cause the station to go to a new setup if there is no priority lot for the current setup even though there might be a non-priority lot waiting. This rule will be refer to as PR_SSU because priority (PR) is given preference over same setup (SSU).

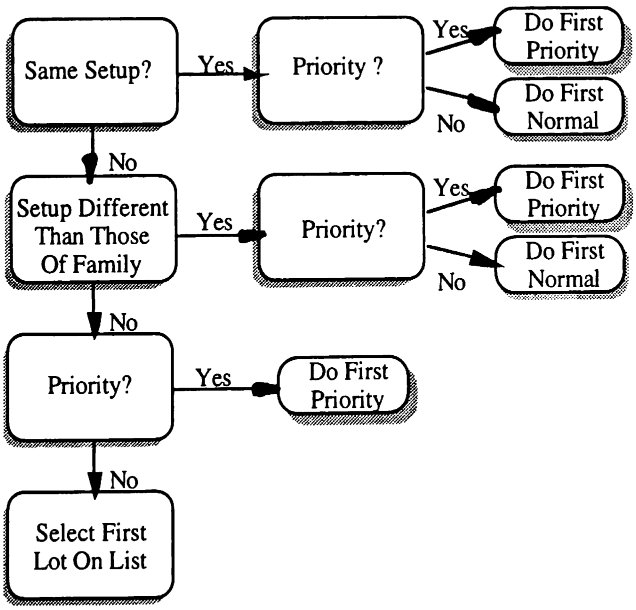

Figure 6: Same Setup & Priority

Another variation of this combination is to give preference to priority lots. The following flow chart will cause the station to go to a new setup if there is no priority lot for the current setup even though there might be a non-priority lot with the same setup waiting. This rule will be referred to as PR_SSU because priority (PR) is given preference over same setup (SSU).

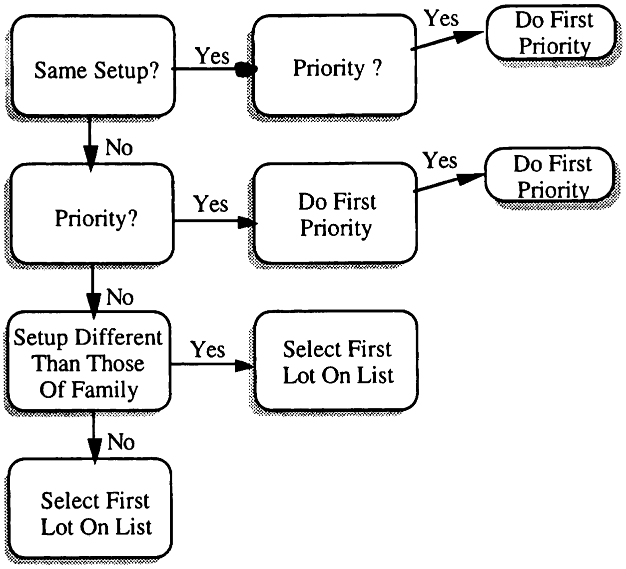

Figure 7: Priority & Same Setup

The results of these two rules show how the factory performance can be fine-tuned with the rules. The following indicate the performance for the two rules.

	FIFO	SSU	PRIOR	SSU_PR	PR_SSU
Non priority lots:					
Avg cycle time days	2.58	2.16	2.69	2.12	2.52
Lots completed	165	192	136	184	158
Max cycle time days	3.32	2.98	3.65	4.32	4.18
Priority lots					
Avg cycle time days	2.53	2.15	1.6	1.99	1.59
Lots completed	36	40	63	44	65
Max cycle time days	3.27	2.88	1.98	4.31	2.03
Total lots	201	232	199	228	223

It is interesting that the total lots produced is about the same. Giving preference to priority yields slightly less in production but significantly more than a "priority only" rule. Further investigation might include adding a due date or value component to the rule. It should be noted that for the combination rules were applied to all non-batching processes — the cure ovens used same setup. Scheduling of batching processes have several additional dimensions which need to be considered and are worthy of a study of their own.

Which rule is best? The decision is management's to make. The management's goals and objectives will guide the selection of the appropriate rule.

SUMMARY

The intent of this paper is not to propose a universal truth in the application of scheduling rules. In the past, research has shown that shortest processing time is the most efficient way to sequence lots. These studies used a hypothetical model that did not require setup times. The model in this paper included setup times, batching, and repeated processes. The point is that the results of this study apply only to this model. Each manufacturing system is unique and management goals are different. Each scheduling model must be analyzed individually. No off-the-shelf software will fit all scheduling applications: therefore, it is critical to make sure a scheduling system accurately models the system.

This leads to another point where criticism may arise. The model did not include variation in processing and setup times. Scheduling is a process of sequencing only, and does not require and should not include random variation in time. It is recommended that randomness be added to the model so it can be validated against past performance. Thus the effect of randomness on a scheduling model is another subject for study.

The goal of this paper is to introduce the concept of finite capacity scheduling using a discrete event, simulation-based model. The application of various scheduling rules and techniques proved in this case to have a beneficial effect on throughput and cycle time. A methodology for developing scheduling rules might have started to surface. The hope is that the reader will receive a glimpse of the power simulation-based scheduling can give to the manufacturing community.

REFERENCES

Thompson, M.B., 1990, "Simulation Based Scheduling" Technical Report # MS90 - 348. Society of Manufacturing Engineers, presented at the International Manufacturing Technology Conference (Chicago, Illinois, Sept. 7).

Rickel, J., 1988 "Issues in the Design of Scheduling Systems," Expert Systems and Intelligent Manufacturing - Michael D. Oliff Editor. Elsevier Science Publishing Co., Inc., New York, New York.

Conway, R. W., Maxwell, 1967 W. L., and Miller, L. W., Theory of Scheduling. Addison-Wesley, Reading, Mass.

Norman, T. A., 1989 "Tailoring Dispatching Rules for Job Shop Scheduling," A White Paper Developed in the Department of Computer Science at Brigham Young University, Provo, Utah.

Thompson, M.B., 1992, "AutoSched Finite Capacity Scheduling and Planning" A White Paper.

Nichols, J. C., Thompson, M.B., 1993, "The Application of Finite Capacity Planning and Scheduling to Semiconductor Testing and Finishing Organizations," first presented at Semicon West '93 sponsored by SEMI (Semiconductor Equipment and Materials International)

Goldratt, Eliyahu M. and Fox, Robert E. *The Race,* published by North River Press. Inc.

BIOGRAPHY

JOHN EVERTON is a member of the society of Manufacturing Engineers, American Production and Inventory Control Society (CPIM), and International Institute of Industrial Engineers. He has authored papers on the application of simulation in material handling, warehousing and manufacturing systems. He is a scheduling consultant at AutoSimulations. He was previously employed by Eaton-Kenway, where he was responsible for the design of automated material handling systems. He has been involved with modeling manufacturing systems for over 16 years. He holds B.S. Degree in Industrial Engineering from the University of Utah.