SIMULATION-BASED ONLINE SCHEDULING IN A MAKE-TO-ORDER JOB SHOP

D. Krige[1], **J. Bekker**[2]* and **C. Schutte**[3]

[1,2,3]Department of Industrial Engineering
University of Stellenbosch, South Africa
[1]david.krige@justenough.com, [2*]jb2@sun.ac.za, [3]corne@sun.ac.za

ABSTRACT

Scheduling is a core activity in the manufacturing business. It assists with the efficient and effective utilisation of capital-intensive resources and increased throughput, thus increasing profitability. Simulation is appealing in manufacturing, as it can realistically imitate dynamic, stochastic processes while being descriptive in predicting the future process. We combined simulation and scheduling and developed an online simulation-based scheduler for manufacturing orders in a South African make-to-order job shop enterprise. There are frequent changes in this type of environment, including random arrivals of orders with stochastic processing times. A simulation-based scheduler is applicable in this myopic, stochastic environment, and we demonstrate its use under these conditions.

OPSOMMING

Skedulering is ’n kern-aktiwiteit in ’n vervaardigingsonderneming. Dit ondersteun doeltreffende en effektiewe benutting van kapitaal-intensiewe hulpbronne asook verhoogde produksiedeurset, wat weer wins verhoog. Simulasie is van nut in vervaardiging omdat dit dinamiese, stogastiese prosesse realisties kan naboots terwyl dit die prosestoekoms op beskrywende wyse toon. Simulasie en skedulering is gekombineer in hierdie projek om ’n simulasiegebaseerde skeduleerder te ontwikkel vir bestellings in ’n maak-op-aanvraag werkwinkel. Veranderings vind gereeld in hierdie tipe omgewing plaas, en sluit toevallige aankomste van bestellings met stogastiese prosestye in. ’n Simulasiegebaseerde skeduleerder is toepaslik in hierdie stogastiese, korttermyn-omgewing, en die werking van die skeduleerder word in hierdie omstandighede gedemonstreer.

[1] The author was enrolled for an MSc Eng (Industrial) degree in the Department of Industrial Engineering, University of Stellenbosch.

[2] Corresponding author

1. INTRODUCTION

In manufacturing, it is widely accepted that producing as few product variants as possible is the most cost-effective business approach. However, there will always be a need for low volume, once-off engineering parts. These once-off parts are usually produced in a make-to-order job shop. This type of job shop has an unpredictable production environment: future order arrivals are unknown, while system disturbances often occur due to machine failures and employee absenteeism. Determining a schedule in such a dynamic environment is challenging because time-consuming rescheduling has to be carried out frequently.

In this paper we describe a scheduler that was developed using discrete-event simulation with online scheduling capabilities, supported by a web-based information system, for a South African make-to-order job shop. The scheduler considers the status of the shop floor each time a new order is received; the current schedule is reviewed, and may be revised at that point. Several classic scheduling dispatching rules and performance measures were incorporated into the scheduler. Dispatching rules include First-in-First-out (FIFO), Earliest Due Date (EDD), Longest Processing Time (LPT), Shortest Processing Time (SPT), Smallest Slack (SS), and Critical Ratio (CR) (also see Shnits et al.[1] and Montazeri & Van Wassenhove [2]). Performance measures included are Makespan, Total Completion Time, Earliness, Lateness, Average Flow Time, and Machine Usage (Shnits et al. [1], Leung [3]).

The industry partner in this study was Daliff Engineering, situated in Airport Industria, Cape Town. It is a manufacturing job shop that produces custom-designed, high-precision parts mainly for aerospace applications. Orders arrive randomly, and usually require the production of small quantities of parts that are seldom reproduced in future. Daliff Engineering is thus a make-to-order job shop, and most of the work is done in the Computer Numerical Control (CNC) machine section, which consists of 13 machines. This study focused on the CNC section only.

The proposed scheduler was evaluated to determine its worth in the application job shop, using test data and designed confidence-building tests. Its performance was also compared with an actual, historical schedule; and since it performed satisfactorily, it will be implemented as the final phase of this project.

We give a brief overview of scheduling below, followed by an outline of the job shop and its scheduling problem. We explain the applicability of discrete-event simulation in scheduling, and present the architecture of the simulation-based scheduler that was developed. Its evaluation is also discussed. Its functioning was demonstrated in a stochastic environment using a secondary simulation model of the job shop. The results of these evaluations and demonstrations are presented and conclusions are offered.

2. SCHEDULING IN MANUFACTURING SYSTEMS: A BRIEF OVERVIEW

Baker [4] defined scheduling as the “allocation of resources over time to perform a collection of tasks”. Pezzella et al. [5] state that scheduling is one of the most critical concerns in the planning and managing of manufacturing processes.

An *order* in a manufacturing system usually requires that one or more *operations* be executed by limited *resources* (e.g. machines), while adhering to a *sequence* or *precedence* constraint. A resource may be considered for several operations. The final plan stating which operation is allocated to which resource, its sequence and time duration, is the *schedule*. The compilation of the schedule is driven by the dispatching rules and performance criteria listed previously. When a resource becomes free, it has to be decided which of the waiting operations (if there are any in the queue awaiting the resource) is to be processed next. To make this decision, the dispatch rule is applied.

The quantity of information known when scheduling begins determines three classes of scheduling: 1) Offline deterministic scheduling, 2) Stochastic scheduling, and 3) Online deterministic scheduling (Pinedo in Leung [6]). Offline deterministic scheduling is applicable in manufacturing systems if all information is known when a schedule is developed. There are no system disturbances and few production changes, because operations are computer-controlled, setup and processing times are deterministic and known, and all the production information – the number of jobs, their release and due dates, production plans, etc – is known. Combinatorial optimisation techniques are used to determine the best schedule that minimises/maximises an objective function.

In stochastic scheduling, the arrival pattern of orders (requiring operations) is assumed to be known, but the processing times of operations are stochastic, so some information is known and some is distributional.

In online deterministic scheduling, no information is known at the time the decision maker has to determine a schedule, and an objective function must be optimised with little information about the future. Since the decision-maker can only determine the best action to take every time a new order is released, an objective function value results that is worse compared with the same situation in the offline case. The online case can be further divided into two subclasses: the processing time of an operation becomes known when 1) the operation begins, or 2) the operation finishes.

Dynamic scheduling can be done in both offline and online scheduling. Many manufacturing processes require dynamic scheduling as orders arrive and machines break down over time. Dynamic scheduling is defined by Church and Uzsoy [7] as "scheduling that aims to update an existing schedule by reacting to the occurrence of *n* predictable events". Artigues et al. [8] state that there are two types of dynamic scheduling: incremental and regenerative. Incremental scheduling leaves the currently scheduled operations as they are, and adds the schedule for the new operations to the existing schedule. With the existing operations taking priority, the lead-time of the new operations may be very long. Regenerative scheduling generates a new schedule for all the operations of new and existing orders. Operations that have already started are not included in regenerative scheduling.

The quality of a developed schedule is measured using certain performance criteria. The performance criteria translate the objectives of the system, and could be customer- or system-oriented. Customer-oriented performance criteria ensure customer satisfaction and a good level of service. A typical objective is to minimise late deliveries. System-oriented performance criteria address system performance, such as minimising work in progress and flow time. A change in external conditions (such as a change in market demands, organisation objectives, the priority of orders, etc) affects the system objectives as expressed by the scheduling criteria. A change of internal conditions (such as delays on the shop floor or machine breakdowns) affects part routing, dispatching rules, delivery dates, and other control decisions.

We now discuss the specific scheduling environment of our focus, the manufacturing job shop.

3. THE JOB SHOP

Hopp and Spearman [9] define a job shop as a place where "small lots are produced with high variety of routings through the plant. Flow through the plant is jumbled, setups are common, and the environment has more of an atmosphere of project work than pacing". In general, job shops specialise in a particular field that requires special skills.

Although there is a close relationship between job, flow, and open shops, Pinedo [6] clarifies the differences. A job shop has fixed routes for jobs, which differ from job to job. In a flow shop the routes of jobs are fixed and remain the same for each job. In a flow shop the machines are set up in series and the jobs flow in the same direction through the series. Jobs in a job shop follow the routes assigned to them and visit resources in different sequences. Open shops have machines that can do all the operations. The routes are thus not fixed and are not defined according to the job. The flow of jobs is thus dynamic and adjusted to suit a schedule.

Figure 1 shows which type of shop is applicable in different production environments. The flow shop is best suited when mass production of common parts is needed. When a great variety of customised parts, but only a few of each, must be manufactured, a job shop is applicable.

The size of orders and the type of parts of an order determine the type of job shop. One such type of job shop is the make-to-order job shop in which orders drive the production, where part routing and processing times are determined by the orders. The order sizes are typically of small quantities, and the parts are custom-designed and are seldom manufactured again. Daliff Engineering, the industry partner in this study, is a typical make-to-order job shop.

The job-shop scheduling problem is now formally presented.

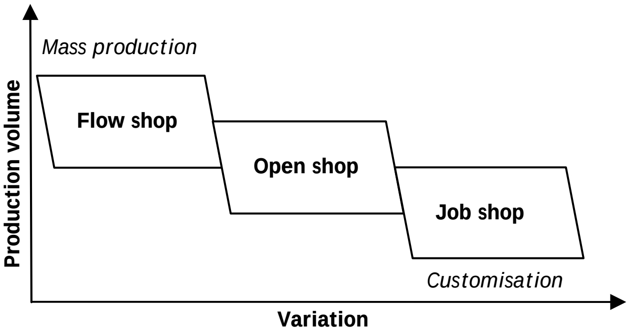

Figure 1. Type of shop according to manufacturing environment

4. THE JOB SHOP SCHEDULING PROBLEM

A mathematical description of the job shop scheduling problem (JSP), as defined by Leung,[3], is as follows:

- A set J of n jobs $J_1, J_2, \ldots, J_n$ has to be processed on a set M of m different machines $M_1, M_2, \ldots, M_m$.
- Each job J_j consists of a sequence of i_j operations $O_{1,j}, O_{2,j}, \ldots, O_{i_j,j}$ that must be scheduled in this order.
- An operation needs only to be processed on a specific machine among the m available ones.
- Pre-emption is not allowed, and machines can handle one operation at a time.
- Operation $O_{i,j}$ has a fixed processing time $p_{i,j}$.
- The objective is to find an operating sequence for each machine that meets a stated performance criterion - for example, minimise the makespan, where the makespan is $c_{\max} = \max_{j=1,n} c_j$, and c_j denotes the completion time of the last operation of job J_j, $(j = 1, \ldots, n)$.

According to Sadeh [10], job-shop scheduling is a Constraint Satisfaction Problem (CSP) or Constraint Optimisation Problem (COP). The constraints that must be satisfied are those of precedence, capacity, release dates, and due dates. The precedence constraints ensure that the job follows the process route assigned to it. The capacity constraints prevent allocation of multiple operations to the same resource at the same time. The date-related constraints determine the possible time frame in which a job can be executed. It is possible that the due date constraint is not met, but some sort of penalty will then occur.

Sadeh [10] uses the schematic in Figure 2 to explain the job shop scheduling problem. In this problem, there are four jobs on five machines. Each node in the figure represents an operation and is labelled with the name of the operation ($O_{i,j}$), where i is the i-th operation and j is the j-th part; the k-th resource required is indicated by R_k. The duration of the operation is simply shown by a number d. The arrows represent the precedence constraints and the broken lines the capacity constraints. This example assumes that each resource can only do one operation at a time, hence the capacity constraint. If more than one operation is competing for a resource, all but one have to wait, as they cannot be processed at the same time.

The constraints can be described by referring to Figure 2. Operation $O_{1,3}$ has to be performed before operations $O_{2,3}$ and $O_{3,3}$, hence the precedence constraint. Operations $O_{1,1}$, $O_{1,2}$ and $O_{2,3}$ all have to be performed on resource R_1, hence the capacity constraint.

Sadeh further states that when some solutions are preferred to others, the job-shop problem becomes a COP with an objective function to optimise. Several scheduling performance criteria exist, each applicable in different scheduling domains.

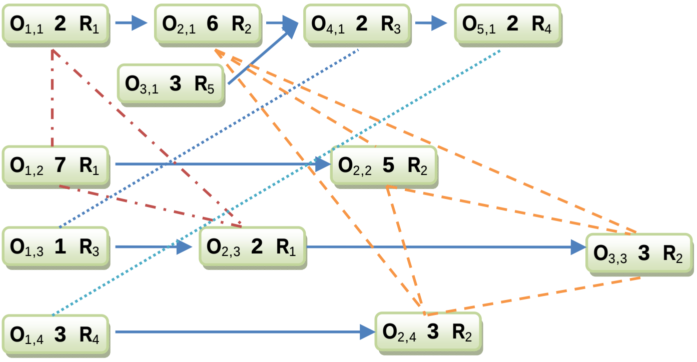

Figure 2. Simple job shop problem with four jobs (Sadeh [10])

It is well known that the job shop scheduling problem is very hard to solve (see Leung [3]), and current systems that have been developed can only solve problems with fewer than 200 operations (Perregaard & Clausen [11]). Another example of the complexity is the classic job shop problem created by Fisher and Thompson [12] with 10 jobs and 10 machines, which took more than 25 years to solve (Schutten [13]). Hopp and Spearman [9] indicate that for the 10-job 10-machine problem there are almost 4 x 10^{65} possible schedules.

The JSP is the most difficult problem in the area of scheduling, according to Pezzella et al. [5]. As the JSP has attracted a considerable amount of research, many techniques – of which the branch-and-bound method and its variations seem to be the most popular – have been developed to solve the problem. In this study, the application of simulation was investigated as a possible scheduling method, and will be discussed next.

5. DISCRETE-EVENT SIMULATION AND SCHEDULING

Discrete-event simulation (DES) can be described as the imitation of the operation of a real-world process or system in which the system state changes at discrete, and possibly random, points in time (Schriber & Brunner [14]). These characteristics enable the use of simulation in an online scheduler.

During operation of a make-to-order manufacturing process the system state changes frequently, mainly due to the unexpected arrivals of new orders, which could enforce rescheduling. Also, setup and processing times are usually unknown and can at best be guessed at, based on experience. This makes the scheduling problem *online*, *dynamic*, and *stochastic*, according to the discussion in Section 2, and discrete-event simulation is applicable to such problems. The current shop-floor status, current orders, and new orders can be considered with a scheduler using a simulation model of the shop floor. Schedules using different dispatching rules can be generated using different performance criteria, i.e. the simulation model takes the current shop-floor status as a point of departure, and predicts the future using the estimated times.

Kim [15] developed such a scheduler, containing a simulation model and a real-time control system, based on the scheduling/ rescheduling approach. The simulation model evaluates various dispatching rules, and selects the best one for a given performance criterion. The rules that were selected are the input to the control system. The real-time control system periodically monitors the shop floor and checks the system performance value. A new simulation is executed when the performance of the system significantly differs from the predicted behaviour, or when there is a major disturbance in the system. If a machine fails and has to be repaired, a new simulation is run to determine a new schedule without the machine until it is fixed, when another new simulation will be run.

6. ARCHITECTURE OF THE SIMULATION-BASED SCHEDULER

An architecture was developed for the proposed scheduler for Daliff Engineering, shown in Figure 3. The input of the scheduler has two components: the enterprise information system, and the shop floor. The information system provides information on the orders that the user inputs. Each order consists of one or more parts, and a part requires various machining operations, e.g. milling of a face or cutting a recess.

For each operation, the manufacturing planner enters estimated times for setup and machining durations. These times can be provided either with minimum and maximum estimations, resulting in continuous uniform distributions, or with minimum, most likely, and maximum estimations, resulting in triangular distributions. These distributions add to the stochastic nature of the scheduling process. When scheduling, the means of the distributions are used. For example, when a schedule is developed according to the Shortest Processing Time (SPT) rule, the operations with distributions that have the shortest expected processing times are scheduled first. However, during the simulation runs, random values for the processing times are drawn from the specified distributions.

The shop-floor component indicates the current state of the shop floor, and these inputs drive the second component of the architecture – the simulation model. Representing the enterprise machine setup and flow of jobs, this model is configured according to the inputs to represent the current shop-floor and order status. The configuration action effectively rearranges waiting operations in each queue of each machine, according to the dispatch rule under consideration. This approach makes rescheduling less complicated and thus much faster, compared with the exact methods mentioned in Section 3, allowing practical implementation of the scheduler. Schedules cannot be guaranteed to be optimal, but it is unnecessary effort to continue searching for optimality when the job pool changes frequently. When the configuration is completed, the simulation model considers the operations per order to be processed, and estimates the performance of the different scheduling rules by doing different simulation runs (scenarios).

The performance of each scenario is recorded for analysis. The scenarios are compared to determine which scheduling rule must be implemented, based on the performance criteria of the scenarios selected by the user. If the user wants to decrease the makespan of the orders, the scheduling rule that results in the shortest processing time of the current list of orders will be chosen as the best one. The scheduler produces an updated schedule that can be followed as the user chooses.

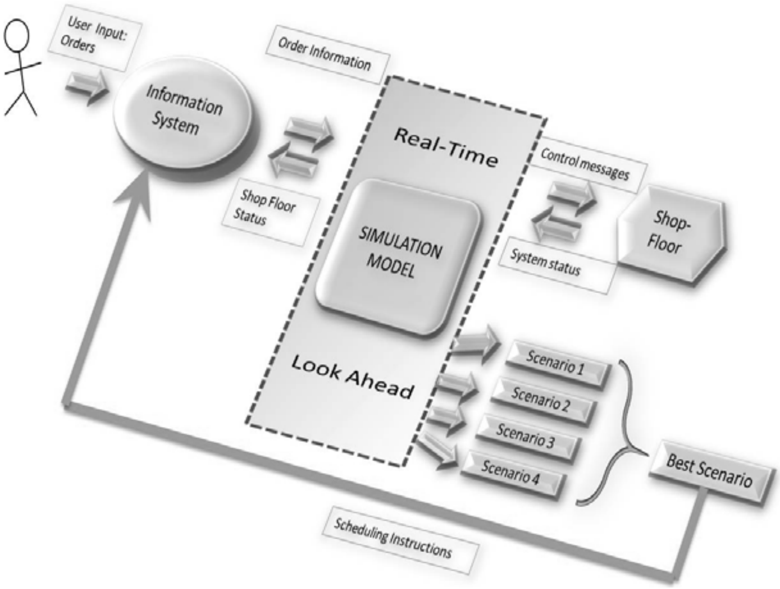

Figure 3. Top-level architecture of the scheduler

7. IMPLEMENTATION OF THE ARCHITECTURE

The information system has been developed to act as an input platform for the simulation model. It is a web-based information system developed in MS FrontPage using ASP coding for dynamic functioning, while the data structure was implemented in MS Access. The information system enables the user to add new orders, customers, and materials. It also has the capability to give the user summary reports of the current quotes, orders and operations, and completed orders.

The information system enables the production planner to generate a quote electronically, which is automatically stored in the database. When the customer accepts the quote, the production planner changes the quote to an order using the information system. The information system then implements this change in quote status and configures the order information to become input to the simulation model.

The simulation model is implemented in the simulation software Arena (Rockwell Software [16]). This software was chosen because it accommodates the discrete, stochastic nature of the system under study, and allows for customisation through Visual Basic for Applications (VBA) on the Microsoft platform. The simulation model was implemented as two components: the Arena Model and the VBA code. The Arena model represents the configuration of the shop floor, while the VBA code handles the customisation of the order configuration according to the current state of the system and the selected dispatching rule.

The model output file is an MS Excel workbook, which has a worksheet for each type of scheduling rule and a worksheet for result comparison. The attribute values for each manufacturing task (or operation) are written to the appropriate worksheet by the simulation model. The structure of the attributes can be seen in Table 1. Each operation has an identification number (Part_ID), operation number (OpsNo), Planned start time, Planned end time, and machine identification number (Machine ID).

Part_ID	OpsNo	Start Time	End Time	Machine ID
2	7	45	55	4
6	16	62	73	5

Table 1: Structure of recorded operation information

The duration of the operations is calculated, and a detailed schedule and a bar chart are constructed from this data. An example of a bar chart is shown in Figure 4, for the CR (Critical Ratio) rule. (The CR is the ratio of the remaining process time and the time to the due date; see Schnits et al. [1].)

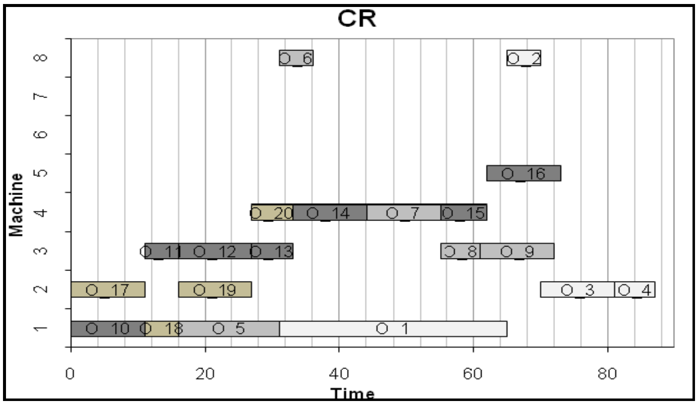

Figure 4: Example of a schedule developed in the output file

The information about each part is also written to the worksheet, i.e. the Part_ID, its Due date, Process end time, Hours late, and Hours early. The makespan, total earliness, total lateness, and average flow time of the schedule, and the average usage of each machine, are also recorded.

The comparison worksheet proposes a schedule under each scheduling rule. Bar charts are compiled to compare the different results, making it possible to compare scheduling rule performances visually per measuring criterion. Figure 5 to Figure 9 illustrate the comparison bar charts of the performance measures.

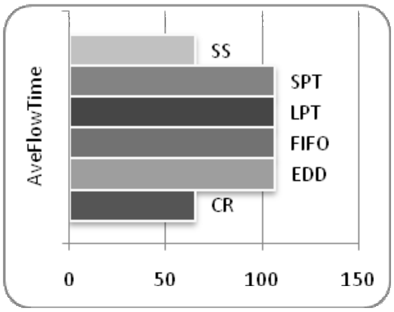

Figure 5: Typical average flow time comparison bar chart

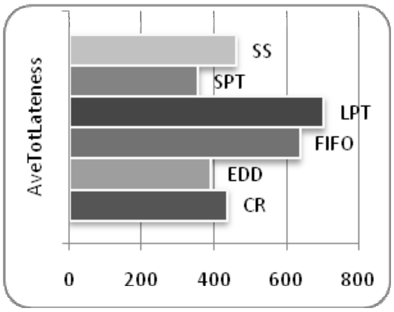

Figure 6: Typical average total lateness comparison bar chart

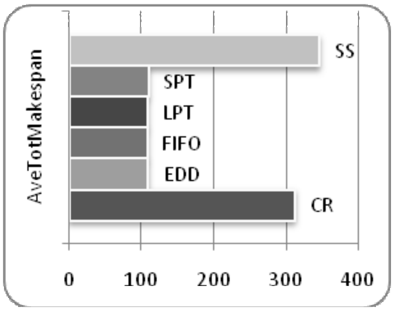

Figure 7: Typical average total makespan comparison bar chart

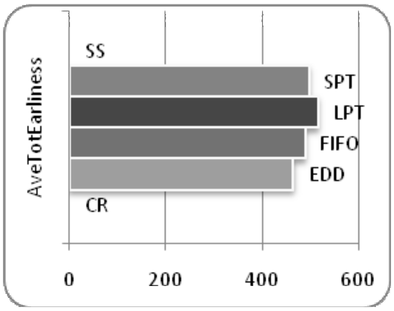

Figure 8: Typical average total earliness comparison bar chart

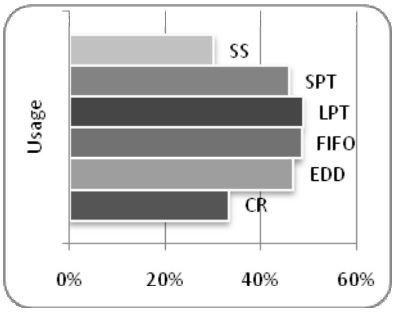

Figure 9: Typical usage comparison bar chart

8. EVALUATION OF THE SCHEDULER

The scheduler needed evaluation to confirm that it functioned correctly. Simple tests of small scheduling problems whose outcomes could be determined were designed and executed, and it was found that the scheduler worked correctly. In order to develop further confidence in the functionality of the scheduler, the performance of its proposed schedules was compared with the performance of the actual schedule that was followed at Daliff Engineering over a two-week period. All the orders that were actually processed during this period were provided to the scheduler.

A summary of the comparison between the proposed schedules and the actual schedule is shown in Table 2. The comparison is made in terms of the completion times of the parts, and not the individual operational times of the parts, because the delivery of parts is the final outcome of the manufacturing process.

Table 2 has five columns, each representing a comparison criterion. The first column shows the total hours gained or lost per scheduling rule, in terms of parts being finished earlier than parts in the actual schedule. The second column shows the average hours gained or lost per part. The third column shows the total number of parts that were finished earlier than the parts in the actual schedule, whilst the fourth column shows the number of parts that were finished later. The fifth column shows the percentage of cumulative hours by which a scheduling rule delivers parts earlier, relative to the historical schedule.

From the table it is evident that the proposed schedule developed under the SPT rule has the best result and would, if implemented instead of the actual schedule, have produced the parts 4.34 hours earlier on average. The proposed schedule of the LPT rule has the worst result, seeing that on average parts are delivered later than by the actual schedule.

There are processing periods that overlap in these schedules because some of the machines have a capacity of more than one. This enables the scheduler to place more than one operation on a logical machine name at the same time, but the name only refers to a collection of similar physical machines. Machines two, seven, and eight have a capacity of two, two, and four respectively, and the rest of the machines a capacity of only one. A detailed list, having the start and end time of each operation, is developed by the scheduler, and it can be used to see how operations were scheduled on these machines.

Scheduling rule	Total hours earlier	Hours per part earlier	Number of parts earlier	Number of parts later	Percentage of hours earlier
FIFO	173.17	2.94	28	31	17%
LPT	-102.08	-1.73	28	31	-10%
SPT	256.17	4.34	36	23	24%
EDD	202.67	3.44	30	29	19%
SS	245.67	4.16	36	23	23%
CR	3.17	0.05	27	32	0%
Mixed	189.67	3.22	28	31	18%

Note:

FIFO	= First-in-First-out	EDD	= Earliest Due Date
LPT	= Longest Processing Time	SS	= Smallest Slack
SPT	= Shortest Processing Time	CR	= Critical Ratio

Table 2: Summary of comparison of schedules

The actual schedule for the evaluation period is shown in Figure 10, and the proposed schedule developed under the shortest processing time (SPT) rule in Figure 11. The proposed SPT schedule on average delivered parts 4.34 hours per part earlier than the actual schedule did. The proposed schedule is less fragmented (see Figure 10 and Figure 11), which indicates that the stop-start of operations is reduced as a result of improved scheduling. Similar results followed for the rules shown in Table 2. Having developed confidence in the scheduler, it was finally tested in a stochastic environment.

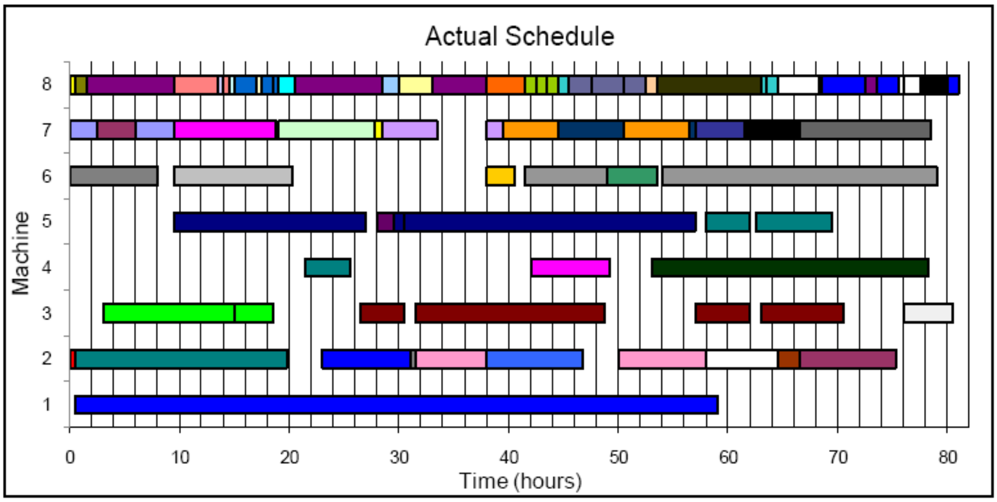

Figure 10: Actual Schedule

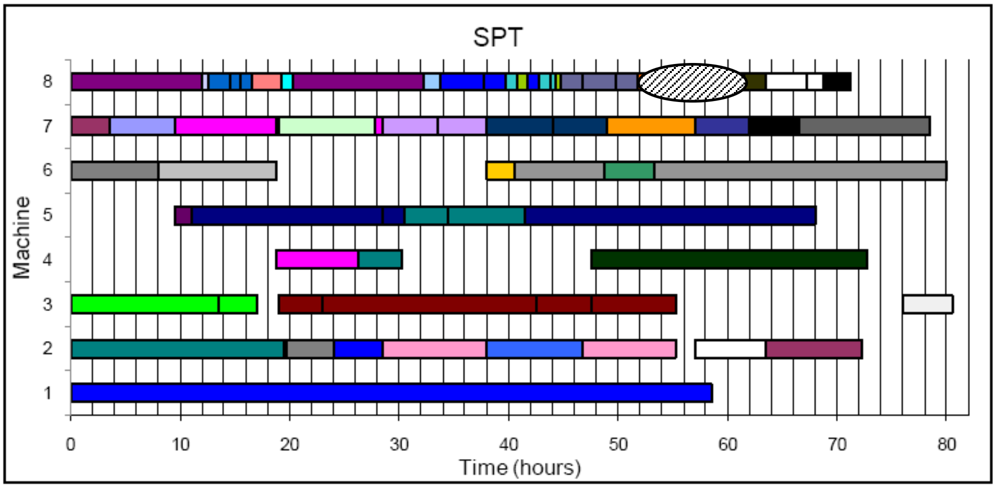

Figure 11: Proposed Schedule under the SPT rule

9. THE SCHEDULER IN A STOCHASTIC ENVIRONMENT

The final step in the development of the scheduler was to demonstrate its functioning in a stochastic environment, before implementing it. We emulated the real-world shop floor as it processed orders over time using a *separate* simulation model (the shop-floor simulation model - SFSM), and created order arrival events at arbitrarily chosen times in that model. At these times, a varying number of new orders, each containing a number of required operations, was entered into the information system.

The status of the shop floor evolves according to the SFSM, and this status is preserved at each decision epoch, where the scheduler considers existing and new orders. It determines the quality of different dispatching rules using the primary simulation model. The best schedule is then implemented from that point onwards, until the next set of orders arrives. The time-line is shown in Figure 12.

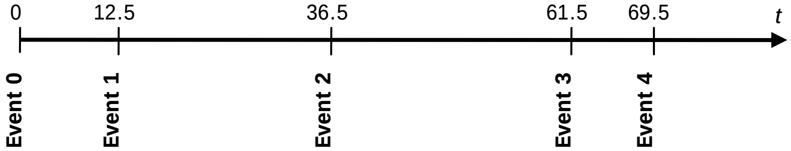

Figure 12: Time-line of random events for scheduler demonstration

The scheduling rule implemented at each rescheduling point is shown in Table 3.

Event number	Time	Rule implemented
0	0.0	FIFO
1	12.5	CR
2	36.5	SPT
3	61.5	CR
4	69.5	CR

Table 3: Rescheduling events and dispatch rules implemented

The online scheduling capability of the scheduling mechanism was confirmed by this demonstration example. In the frequently changing environment where the distant future was unknown, a schedule was developed according to *what was known*, and this schedule was followed until *what was known* was changed by the arrival of new orders. A new schedule was then developed according to the new *what was known* information. Every time a new schedule was developed it was believed to be the best schedule for *what was known*, and it was followed until more was known. This approach is called *myopic scheduling*. In the demonstration, a 'best scheduling rule' was selected via reasoning at each reschedule epoch.

10. CONCLUSION

We proposed a simulation-based scheduler for a make-to-order job shop. An architecture was presented for the scheduler, which comprises an information system, a simulation model, and a real-world shop floor. The scheduler was tested using simple tests with predictable outcomes, followed by an evaluation of its generated schedules compared to the actual schedule followed, and it generally showed improved schedules. The functioning of the scheduler was finally demonstrated using a time-line with random reschedule epochs, and it generated a new schedule at each point, based on finished, existing, and new tasks in the system. The next step is to implement the scheduler for the industrial partner.

The idea of a simulation-based scheduler is not novel, and many similar schedulers have been developed; but we demonstrated the development of such a scheduler in a local make-to-order job shop. Future work will deal with developing a multi-criteria decision support mechanism that will automatically determine and suggest the best schedule to the manufacturing planner.

11. REFERENCES

[1] **Shnits, B., Rubinovitz, J. & Sinreich, D.** 2004. Multi-criteria dynamic scheduling methodology for controlling a flexible manufacturing system. *International Journal of Production Research*, 42, pp. 3457-3472.

[2] **Montazeri, M. & Van Wassenhove, L.N.** 1990. Analysis of scheduling rules for an FMS. *International Journal of Production Research*, 28, pp. 785-802.

[3] **Leung, J.Y-T.** 2004. *Handbook of scheduling: Algorithms, models, and performance analysis*. Chapman & Hall/CRC, New York.

[4] **Baker, K.R.** 1974. *Introduction to sequencing and scheduling*. Wiley & Sons, New Jersey.

[5] **Pezzella, F., Morganti, G. & Ciaschetti, G.** 2008. A genetic algorithm for the Flexible Job-shop Scheduling Problem. *Computers & Operations Research*, 35, pp. 3203–3212.

[6] **Pinedo, M.** 1995. *Scheduling theory, algorithms, and systems*. Prentice Hall, New Jersey.

[7] **Church, L.K. & Uzsoy, R**. 1992. Analysis of periodic and event-driven rescheduling policies in dynamic shops. *International Journal of Computer-Integrated Manufacturing,* 5, pp. 153–163.

[8] **Artigues, C., Michelon, P. & Reusser, S.** 2003. Insertion techniques for static and dynamic resource-constrained project scheduling. *European Journal of Operational Research*, 149(2), pp. 249–267.

[9] **Hopp, W.J. & Spearman, M.L.** 2001. *Factory physics: Foundations of manufacturing management,* 2nd edition. Irwin McGraw-Hill, Boston.

[10] **Sadeh, N.** 1991. Look-ahead techniques for micro-opportunistic job-shop scheduling [Doctoral Thesis]. Carnegie Mello University, Pennsylvania.

[11] **Perregaard, M. & Clausen, J.** 1998. Parallel branch-and-bound methods for the job-shop scheduling problem. *Annals of Operations Research*, 83, pp. 137–160.

[12] **Fisher, H. & Thompson, G.L.** 1963. Probabilistic learning combinations of local job-shop scheduling rules. *Industrial Scheduling,* Prentice-Hall, Englewood Cliffs, pp. 225–241.

[13] **Schutten, J.M.J.** 1998. Practical job-shop scheduling. *Annals of Operations Research,* 83, pp. 161–177.

[14] **Schriber, T.J. & Brunner, D.T.** 2005. Inside discrete-event simulation software: How it works and why it matters, in Proceedings of the 2005 Winter Simulation Conference, M.E. Kuhl, N.M. Steiger, F.B. Armstrong & J.A. Joines (eds), http://www.wintersim.org/prog05.htm, cited December 2008.

[15] **Kim, M.H.** 1994. Simulation-based real-time scheduling in a flexible manufacturing system. *Journal of Manufacturing Systems,* 13(2), pp. 85-93.

[16] **Rockwell Software**. 2008. www.arenasimulation.com.