

Experiments with a knowledge-based simulator and scheduler system

G. L. Kovács

Computer and Automation Research Institute, Hungarian Academy of Sciences

H-1111 Budapest, Kende u. 13-17., Hungary,

Fax: 36-1-1667-503; E-mail: h673kov@ella.hu

Abstract

Computer aided simulation can assist in Flexible Manufacturing Systems' (FMS) design (planning), scheduling and operation as well. The knowledge for planning and controlling of FMS can be built into knowledge based (KB) systems. KB, intelligent control of FMS can be done by using the results of scheduling (production planning based on process plans). Intelligent control and supervision of a system can be solved by real-time, intelligent environments. In this paper two KB simulation-scheduler systems will be discussed. In the first one the simulation system is implemented in SIMAN/Cinema, while in the second one in G2. The simulation is connected to preparation, evaluation and scheduling advisory systems implemented in G2 in both systems. The paper gives details on scheduling results using both systems with the same two test cases.

Keywords

Simulation, scheduling, knowledge-based system, flexible manufacturing

1 INTRODUCTION

The production process in a manufacturing system is based on the manufacturing schedule which is created from a process plan. A process plan consists of the operation order, the identifiers of the machines, tools, etc. and the time period of each operation on the given manufacturing equipment. The manufacturing schedule contains the previous information plus the start and end times of all the

operations. A good scheduler can take into account the transportation and set-up times as well. A scheduler also has to consider the due dates (or due times). There are different solutions to obtain appropriate production schedules. Some of them are in the following categories: mathematical programming, multi-criterion decision making, heuristic approach, graph-theoretical approach, application of simulation, application of AI tools.

Our approach is a combination of the last two methods, when a simulation system will be connected to a KB scheduler.

2 APPLICATION OF KB SIMULATION FOR SCHEDULING

Scheduling with simulation is usually fast and reliable. During the simulated manufacturing process the starting and finishing time of each operation can be recorded, and later on a schedule can be built up based on these information.

If we create a manufacturing schedule with simulation, there are several decision points - even if only simple algorithms are used - in the process of the simulation. For example:

- if there are more than one workpieces in a parallel buffer (temporary storage), the question is which workpiece can leave the buffer first, or
- more than one machine tools are suitable for a certain operation, which machine to choose,
- more than one workpieces are waiting for the same resource, which will be served first, - etc.

In traditional languages and in most of the simulation languages these kind of decisions can be made by using the if-then-else structure only. It is rather difficult to write decision rules if we have a large number of conditions to be satisfied at the same time. Using the traditional languages even not too complicated scheduling rules as LIFO or FIFO are hard to manage. Combining simulation and knowledge based (KB) systems will result in the so called knowledge based simulation to solve the problem of decision making during simulation. In such a combination the scheduling power will be defined by the KB.

3 KB PROTOTYPE FMS SIMULATION-SCHEDULER SYSTEMS

Our first KB FMS Simulation system was written in CS-PROLOG (Communicating Sequential Prolog) [1]. The next version was based on the traditional SIMAN/Cinema simulation-animation [2] package connected with advisory systems implemented in the ALL-EX [3] expert system shell. Then ALL-EX was substituted by the G2, real-time, object oriented expert system environment [4] (Hybrid System). Finally the simulation was implemented in G2 as well, this way the same object definitions and some part of the knowledge could be used for all subsystems and there were no communication problems between the simulation and the KB (KB System). [5] gives a detailed description of system

development and implementation problems including some comparisons of the above KB simulation-scheduler systems.

3.1 Simulation and advisory systems in the same G2 environment

The natural advantage of the system where the simulation and the interconnected expert systems (preparation, evaluation and scheduler) are implemented in the same G2 environment is that neither interfacing nor co-operation problems appeared as in the previous prototypes, where G2 had to cooperate with SIMAN/Cinema. The biggest difference was caused by the G2's object-oriented view. All elements of the simulated system are now represented by G2 objects with a hierarchy between them to make attribute inheritance possible.

4 EXPERIMENTS USING THE G2 ENVIRONMENT

[6] gives a good summary of scheduling (and dispatching or priority) rules which are often applied in experimental systems. There is a distinction between dispatching (or priority) and scheduling rules, which is the following according to a simplified definition: Dispatching rules assign scalar values to all waiting jobs, while scheduling rules select the job to be scheduled next in preference to the other waiting jobs. However in most practical implementations priority rules are used for both purposes. In the following we shall use only the scheduling rules for both terms, and we do not distinguish between pre-release decisions, part-release decisions and part-routing decisions which are the essence of dispatching.

Some of the possible scheduling priorities and rules associated to them are the following:

- Processing time (shortest/longest next operation, shortest/longest remaining time, etc.)
- Due date (earliest, earliest operation, etc.)
- Number of operations (fewest/most remaining, etc.)
- Random arrival time (FIFO, LIFO, random, earliest available, etc.)
- Cost rules (minimum tardiness, etc.)
- Setup time rules (highest/lowest setup time, etc.)
- Slack rules (least amount of slack, least static slack, etc.)
- Machine (shortest processing time, shortest queue behind, etc.)
- Tool (least number of tool movement, etc.)

These priorities and the rules belonging to them freely can be combined in the hope of getting better results. The "better result" means better according to some given goal functions or expectations. Such expectations can be the shortest total production time, cheapest production, uniform load of resources, maximal (or minimal) load of the most expensive/cheapest machine, etc. These can be simple or combined, and they may be contradictory in some cases. And - naturally - they are represented by the applied scheduling algorithm (rules).

Application of expert systems allows to use contradictory rules simultaneously, and probabilities and uncertainties can be applied, too. This way the power of the system can be increased and programming is easier. For our experiments we applied simple rules of different priorities with a goal function of having the shortest throughput time (total production time). The results are evaluated based on total throughput times. Scheduling was divided into two consecutive steps. We separately tried to optimize the sequence, as raw parts arrive into the input buffers of the FMS (pre-schedule), and then optimization is done when the parts are moving in the FMS (schedule).

4.1 Optimizing the input sequence of raw parts arriving into the FMS

In this phase a preliminary sequence of the raw parts is determined how to arrive into the input storage of the FMS to see the effects of these sequences to the final schedule.

The following rules/priorities were used:

- MQ, type of maximum quantity first (always that part type has the highest priority from which the biggest number is required, the basis of comparison is the number of required parts minus number of parts under /plus after/) processing of the given types)
- MC, multicolor input (all types of parts should be in the input storage as long as possible)
- UD, uniform distribution input (the distribution of different parts in the input storage corresponds to the distribution of quantities of different part types to be produced)

4.2 Application of different scheduling algorithms

As it was mentioned, earlier only the well-known Round Robin strategy was built into the SIMAN simulation system itself and a LIFO and a FIFO algorithms were implemented in the Scheduling Advisor first. Later on we changed these algorithms, and for the experiments the following ones were implemented:

- SP, job (part) with shortest process plan first
 - LP, job (part) with longest process plan first
 - SI, job (part) with the shortest next operation time first
 - LI, job (part) with longest next operation time first
 - ED, job (part) with the earliest due date first
 - SR, job with the shortest remaining processing time first
 - LR, job with the longest remaining processing time first
- (Jobs are always associated to parts.)

The time-spans of individual process-plans are measured by generating individual schedules for each part as they were the only single parts in the system.

4.3 Tests with the GoldStar FMS, application of the hybrid system

The goal of the experiments was twofold: to test the hybrid simulation-scheduler and to make test runs with a real industrial example. The FMS works in the workshop of the GoldStar Heavy Industries Cable factory near Seoul. There are 8 CNC machining centers with palette exchanger and tool magazine in the system. The machine tools are situated in one long line, there are seven buffers for temporary palette storage, with 6 slots each. One transport carriage is moving on rails, with a speed of 60 m/min if loaded and with 80 m/min if empty and its accuracy is 0.2 mm. The palette size is 1mx1m, the maximum weight of the workpiece is 3000 kg (3 tons), there is room only for one palette in the carriage. There is a load/unload station with four slots where parts are fixed on the palettes, and where they are unfixd to leave the system. Fixing/defixing is done manually.

To demonstrate the size of the workshop we give the distances of the machine tools from the load/unload station, as machine identifier (distance in meters):

1(12), 2(27), 3(26), 4(33), 5(40), 6(47), 7(54), 8(61)

The production task in a shift (one week) are given in Table 1.

Table 1. Production tasks of the GoldStar system

color	batch size	package size	due date (day)	setup time (min)	operation times (min)		
					op1	op2	sum
red	100	6	6	5	105	77	182
yellow	100	5	6	5	65	50	115
blue	90	10	6	5	30	225	255
brown	30	3	6	5	60	120	180
green	10	2	3	5	70	65	135
white	10	2	3	5	80	65	145
Total	340						

We use the Cinema animation colors instead of the names of the parts to identify them. There are two operations to be done on every part. An operation is defined as all CNC programs on one machine tool with one palette position. The two operations can be done at any machine tools, and there is a need of a new fixing of the part between them.

If we calculate the total amount of time which is theoretically necessary for the production we have:

$$100 \times 182 + 100 \times 115 + 90 \times 255 + 30 \times 180 + 10 \times 135 + 10 \times 145 = 60850$$

If we add two setup times to each workpiece we get:

$$TSUM = 60850 + 340 \times 10 = 64250 \text{ min}$$

TSUM in days is $64250 / 60 / 24 = 44.6$ days if we suppose 1 machine tool only, but as we have 8 similar purpose machine tools

$$TSUM8 = 44.6 / 8 = 5.57 \text{ days}$$

We can say that to produce all parts in a shift 5.57 days were necessary if every machine tool worked full load all the time, i.e. theoretically the production could be done in a working week with 6 days to work with 24 hours work a day. The high amount (42) of buffer places makes such a vision realistic. As the speed of the AGV is high enough to say that the time periods which the palettes spend on the AGV are one-two order smaller than the machining times, transportation times are neglected or we can say that they are included into the setup times.

4.4 Tests on the CIM plant of the TUB, application of the KB program

For these experiments the pure G2 program was used with the goal of checking as many test runs as possible with a complicated, 4-cell FMS, with simple, but realistic production tasks, to find out the differences between the possible combinations of pre-scheduling and scheduling rules. Several different batch sizes of all parts were checked with several scheduling algorithms available (in the Scheduling Advisor system). We made test runs with a real production task where 4 different parts should be produced and checked on the CIM pilot plant of the Technical University of Budapest (TUB). This flexible manufacturing system consists of 4 different cells (manufacturing, storage and AGV, measurement and assembly). The manufacturing cell contains a CNC lathe and a CNC machining center served by two different robots. There is an AGV moving on given traces to carry one palette with one workpiece at a time. All parts have different process plans (Table 2.). The meaning of the abbreviations are the following: fix-fixing, turn-turning, mill-milling, asm-assembly, meas-measurement, dfix-defixing.

Table 2. Process-plans of the four parts of the TUB system

part	operations time (min)	timespan (min)
red	fix(15)-turn(25)-mill(30)-dfix(5)	75
yellow	fix(15)-mill(18)-asm(30)-dfix(5)	68
blue	fix(15)-meas(25)-asm(30)-dfix(5)	75
green	fix(30)-turn(20)-meas(10)-mill(20)-meas(10)-mill(20)-asm(30)-dfix(5)	145

We use the different colors to identify the different parts as it was implemented in the animation of the process. The respective due-date priorities of the part type were given instead of real due dates and they order was the same in all experiments: yellow-green-blue-red. The yellow parts are the most important, the red ones are the least important, if due dates are concerned.

The batch sizes varied from 0 to 30 during the experiments. As the Input Storage of the system has 10 slots the experiments would have been meaningless if the sum of the four batch sizes had not been bigger than 10. We made runs with several combinations of the priorities (3 x 4= 12 runs for each production task).

All the three input sequence algorithms and the following scheduling algorithms were used: LP, SI, LI, ED) with the following 8 sets of batch sizes:

yellow	3	4	5	6	7	8	9	10
green	3	4	5	6	7	8	9	10
blue	3	4	5	6	7	8	9	10
red	3	4	5	6	7	8	9	10

and then with the same numbers, but one part was always missing. As example we give the table without yellow parts.

yellow	0	0	0	0	0	0	0	0
green	3	4	5	6	7	8	9	10
blue	3	4	5	6	7	8	9	10
red	3	4	5	6	7	8	9	10

We had one group of experiments with all part types, one with yellow missing, one with green missing, etc. This way we had 8 times 12=96 runs for each group, i. e. altogether we made 5 times 96 = 480 experimental runs, which took several hours. Comparisons are made only within the same groups, i. e. if the final evaluation will be done we expect results for five distinct types of production, what is naturally far from an exhaustive examination. The results are rather interesting, the statistical evaluation will take some more time, as no direct relationships could yet be determined.

Similar results were gotten with the application of the hybrid system on the same FMS (TUB).

4.5 Some basic features of the test problems

There are some basic differences between the two problems we made experiments with:

- The layout of the TUB system is really two-dimensional, with complicated trace for the AGV to follow.
- The layout of the GoldStar system is one-dimensional, the machine tools are in one row, and there is a parallel row for the buffers, while the AGV moves on rails between the two parallel rows to easily serve both of them.
- The task of the TUB system is more complicated as not only machining, but measurement and assembly functions are done, too, and the system is cell-based.
- The GoldStar system does only machining with equivalent machine-tools
- There is no substitution possibility for any of the resources in the TUB system, thus there cannot be alternative routings for the workpieces.
- There is an almost infinite substitution possibility in the GoldStar system as all machine tools (and all buffers) are of the same type, thus alternative routings can be essential.
- The TUB system has a 10 slot I/O buffer and each resource has a 2 slot buffer for palette exchange.

- The GoldStar system has a 4 slot I/O buffer and each machine tool has a 1 slot buffer, and the system has 42 further buffer storage places for temporary storage of palettes.

Based on the above differences it is clear that we had completely different goals with the two sets of experiments:

- We had the goal of trying to approach to the theoretical value of 5.57 days, which is a lower limit to the given production task in the GoldStar system.

- The goal of the several test-runs on the TUB system was to figure out whether there are relationships between the different production tasks and different pre-scheduling and scheduling algorithms on a given FMS.

4.6 Evaluation of the test-run results

4.6.1 The GoldStar system. For this example we did not deal with the sequence of filling up the 4 slot I/O buffer, but a fixed sequence was used for all test-runs. This was the following: 2xWhite, 2xGreen, 6xBrown, 18xBlue, 20xYellow and 20xRed. This set of 68 workpieces was repeated 5 times. We made runs where the following algorithms (priority rules) were checked: the original embedded scheduling algorithm of SIMAN, and SI, LI, SR, LR, and LP. It is interesting, that the SI and SR rules gave exactly the same results, as well as the application of the LI and LR rules. It was surprising that all runs gave results within the range of 5.6 to 6.1 days, i. e. all were very closed to the theoretical minimum, to the 5.57 days. The differences are so small that no reason to specially evaluate them. But, as we know that the FMS was designed to optimally produce the given parts in the given amounts we can tell that the results are not really surprising. There are two main factors that influence these results: the high speed, and neglected times of the AGV and the high amount (42) of temporary buffers. This way the full load of all machine tools can be reached for all time except of starting and finishing the shift. Similar results were gotten with the application of the KB system on the same FMS (GoldStar).

Now we are in the process of checking whether that FMS is really flexible or not, and we try to make test runs with different production tasks of the same magnitude, as the real one is. The evaluation of the results will take a longer time.

4.6.2 The TUB system. In spite of the very high amount of test-runs we still do not have enough results to have any kind of final conclusions concerning the FMS and the given production tasks on it. Just to give an idea on the measured time values some results were the following (see Table 3.):

Table 3. Some scheduling results of the TUB system

Scheduling rules	workpieces: 5,5,5,5 input sequence	workpieces: 10,10,10,10 input sequence
------------------	---------------------------------------	---

	MQ	UD	MC	MQ	UD	MC
ED	624	638	587	1119	1158	1142
SI	591	653	605	1143	1159	1175
LI	625	622	604	1162	1183	1207
LP	619	625	617	1170	1191	1194

Based on the above and several similar tables we can get only rather general conclusions on the given FMS: generally the ED algorithms are better than the LI ones, there are no significant differences between MQ, UD and MC, etc.

5 SOME REMARKS ON THE APPLICATION OF SIMAN

Since SIMAN is a language developed directly for FMS simulation it has some very pleasant properties which are missing from G2. For instance, in SIMAN it is very convenient to define a transport-device to follow a zigzag route - it is only necessary to define the route in the animation module. Then the transport-device can follow the route only by telling it the start and final station. On the contrary, in G2 a route following algorithm had to be implemented and it did not reach the quality of the SIMAN animation (Cinema), of course it was not our goal, either. Another feature that would be of great use in G2, too, were the presence of predefined statistics and distribution functions. Statistics functions are used in numerical evaluation of the simulation results and distribution functions are required when we wish to randomly generate numbers with a given distribution.

On the other hand program development and model building are definitively easier in G2 than in SIMAN - even for simulation. The object oriented feature of G2 is one of its great advantages.

6 CONCLUSIONS

The implementation of our programs in SIMAN and G2 was fast enough, G2 was excellent to make the different experimental program runs. Data preparation is - however - a tedious and time-consuming activity.

There is a basic problem to be solved: if G2 and SIMAN are used together most data of the production plant, products and of the production have to be duplicated to be there in both systems. This should be avoided by using some kind of automatic data conversion between the object oriented G2 and the traditional SIMAN. This is one of the interesting challenges for our future research and development - if we decide to keep SIMAN/Cinema for simulation and animation. Even if an object-oriented simulation package (e.g. ARENA, the recent version of SIMAN) is used, the problem is not solved automatically.

Based on the test-runs we have to admit that experiments should be designed more carefully, and then more characteristic results can be obtained. In the recent

set of tests we tested our programs, and we can conclude that they can be used for checking different possible production schedules of given FMS. On the other hand the test results are strongly FMS and production task related, so we could not yet find general ideas to suggest for scheduling new tasks. Further experiments should be done on different FMS with different production tasks to be able to define quality differences between the scheduling and dispatching algorithms (rules) discussed in this paper. On the other hand, differences between the pure G2 and the hybrid programs should be further analyzed.

7 ACKNOWLEDGMENTS

This work was partially supported by the Hungarian OTKA (National Scientific Fund) fund, contract No. T014549.

8 REFERENCES

- [1] I. Futó: "AI and Simulation on PROLOG Basis", Proc. of the Int. Symp. on AI, Expert Systems and Languages in Modeling and Simulation, Barcelona, Spain, June 2-4., 1987., pp. 117-124.
- [2] C.D. Pegden, R.E. Shannon, R.P. Sadowski: "Introduction to Simulation Using SIMAN", McGraw-Hill Inc., New York, 1990.
- [3] "ALL-EX Reference Guide", Multilogic Computing Ltd, 1991.
- [4] "G2 Reference Manual", G2 Version 2.0, Gensym Corporation, Cambridge, MA, USA, 1990.
- [5] Kovács, G., Mezgár, I., Nacsa, J., Kopácsi, S., Gavalcová: "A Knowledge Based and a Hybrid Simulation - Scheduling System", Proceedings of the 1994 Japan - USA Symposium on Flexible Automation, Kobe, Japan, July 11-18, 1994, Vol. 2. pp. 783-790.
- [6] Gupta, Y. P. et al: "A Review of Scheduling Rules in Flexible Manufacturing Systems", Int. J. Computer Integrated Manufacturing, 1989, Vol., 2. No. 6, pp. 356-377.