Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

ADAPTIVE FLOW CONTROL IN FLEXIBLE FLOW SHOP PRODUCTION SYSTEMS - A KNOWLEDGE-BASED APPROACH

Mark Aufenanger
Hendrik Varnholt
Wilhelm Dangelmaier

Heinz Nixdorf Institute
Fürstenallee 11, University of Paderborn
33102 Paderborn, Germany

ABSTRACT

Today simulation is essential when researching manufacturing processes or designing production systems. But in the field of manufacturing, simulation can not only be used for purposes of research or design, it can also be utilized by flow control systems in order to make better and faster decisions. In this paper we focus on real-time scheduling in a special kind of flexible flow shop systems. These consist of production stages, which represent groups of machines doing the same work, but working at different speeds. Flow control in these flexible flow shop environments with uniform machines is exceedingly complex and it is even more complex when uncertainties are taken into consideration. For this reason we develop an adaptive scheduling heuristic, utilizing both simulation and artificial intelligence in order to make globally good decisions without causing noticeable manufacturing delays.

1 INTRODUCTION

The more complex a system is, the less predictable is its behavior. For manufacturers this is a dilemma. They produce increasingly sophisticated products, are trade partners in lengthening supply chains, and need to be competitive on a global market. Requirements on production and logistics processes grow. This causes more and more complex plans. At the same time an increasing accuracy should make cost-intensive time and capacity buffers dispensable. Hence, the requirements are in conflict. Meeting them simultaneously can only be successful by utilizing substantially improved planning processes.

In recent decades this did not change despite of an enormous growth of computing power. Production planning is still complex and for this reason time consuming. Solving typical production planning problems often takes hours or even days (cf. Hopp and Spearman (2001)). Hence, most established scheduling methods can only be used if time is not critical. To be able to immediately react to deviations which occur during a running production process other methods are needed.

For this reason, approaches to make planning more flexible moved to the focus of research on production planning. Researchers no longer regard production processes as completely deterministic but as partially stochastic. They realized that most often a production schedule cannot be executed as planned. Consequently researchers had to develop methods being able to quickly react to changes while also being reliable. Such methods are described by Aytug et al. (2005). Among other methods they review techniques which utilize data (knowledge) gathered prior to the production process. By using this knowledge in combination with artificial intelligence algorithms the methods try to predict the outcome of different possible decisions in a given situation.

The idea which Aytug et al. (2005) describe as predictive-reactive lead us to the adaptive scheduling framework which is described in the fourth chapter of this paper. First work on this approach was presented in Aufenanger et al. (2008). In the next section we give an overview of existing research on scheduling in flexible flow shop environments, in particular such approaches dealing with uncertainties in production environments. We conclude with the description of a prototype and present test results.

1.1 The Nature of Flexible Flow Shop Environments

Virtually all research on flexible production planning focuses on job shop scheduling problems or – in fewer cases – flow shop environments. But there are other types of production environments, which are of the same or even greater importance in practice. One of these is the flexible flow shop, which is considered in this paper.

To understand the nature of flexible flow shop environments it is important to first understand flow shops. These consist of different machines, which are arranged in multiple stages. At each stage there is only one machine, and stages have to be passed in a given order. In contrast to this, flexible flow shops have more than one machine at each stage. Hence, in flexible flow shops jobs can pass others.

The machines at each stage of a flexible flow shop can be completely identical: they do the same work at the same speed. Or they can among other possibilities be uniform: they do the same work but at different speeds. In this paper we consider flexible flow shops with uniform machines.

2 STATE OF THE ART IN SCHEDULING FLEXIBLE FLOW SHOPS

2.1 Optimum Techniques

Many researchers who consider scheduling in flexible flow shop environments (Wang 2005) suggest algorithms following the branch-and-bound approach. They argue that there is a more intelligent way than enumerating all possible solutions in order to find an optimal production plan. The branch-and-bound approach (for details cf. for example Dakin (1965) or Brucker (2007)) avoids enumerating all possibilities by calculating upper and lower bounds in order to prune possible solutions which cannot be the optimum as early as possible. Hence, branch-and-bound techniques are methods for implicit enumeration. In practice they are faster than completely enumerating the possible solutions. Nevertheless, their worst-case run-time is not better.

2.2 Heuristics

Utilizing optimal solution techniques turned out to be impractical. Finding the best possible solution is still often a matter of hours or days, even with nowadays computational power. However, an acceptable solution can often be found in seconds when using heuristics (Wang 2005).

A basic heuristic for solving flexible flow shop scheduling problems can be derived from the branch-and-bound approach Blazewicz et al. (2007): If not all generated branches of a decision tree are considered as starting points for a further search for better solutions, the width of the tree and hence the complexity of the search will be significantly reduced. Anyway, the solution quality strongly depends on the "right" selection criteria for branches to explore.

An entire class of heuristics for solving flexible flow shop scheduling problems is based on a different idea, which was first mentioned by Sriskandarajah and Sethi (1989). They divide a flexible flow shop problem in two kinds of sub problems: First, a number of regular flow shop problem is build. Each of the machines on each stage of the flexible flow shop is then assigned to one of these regular flow shops. After that, algorithms which are designed for solving regular flow shop scheduling problems – for example the algorithm of Johnson (1954) (also described in Baker (1974)) – can be applied.

More sophisticated heuristics are needed when considering flexible flow shops with uniform machines. One of the few researchers who developed methods for this kind of problems are Kyparisis and Koulamas (2006). They basically suggest building partial schedules for each production stage by using techniques which are designed for parallel machines problems and then combining these to a complete schedule for the original flexible flow shop problem.

2.3 Artificial Intelligence Techniques

To improve the solution quality heuristics can be enhanced by utilizing artificial intelligence techniques. The basic idea is to imitate the human process of decision making, which is often based on experience. This can be achieved by machine learning methods which create knowledge bases. A knowledge base contains decisions and the observed outcome together with descriptions of the situations the decisions were made in. Scheduling methods can exploit such a knowledge base by comparing the current situation with the situations stored in the knowledge base and determine which decision is the best. Techniques following this approach are usually fast in decision making, but they need extra time prior to the production process to build a knowledge base.

3 DEALING WITH UNCERTAINTIES

Uncertainties in a production environment are as numerous as the influences on the manufacturing process. Machine failures, late material delivery and unavailable operators often make previously generated plans inapplicable and thus useless. For that reason, useful methods for production scheduling need to incorporate uncertainties. Following the taxonomy of Aytug et al. (2005) such approaches can be classified as predictive, reactive or predictive-reactive.

3.1 Predictive Scheduling

Predictive techniques try to anticipate all possible changes to the production parameters. They produce schedules which should be applicable under all circumstances. If for example a processing time is subject to frequent changes, predictive techniques will use the longest possible processing time to build plans. Therefore, the resulting plans are schedules for worst-case scenarios. Relying on them is cost-intensive due to their inefficiency under normal circumstances. For specific methods following the predictive scheduling approach cf. Kouvelis et al. (2000) or Mehta and Uzsoy (1998).

3.2 Reactive Scheduling

Completely reactive approaches on the other hand do not try to build a plan at all. They do not consider scheduling problems from a long term perspective, but make decisions just for the immediate future. These decisions are only affected by local parameters. For example, if currently a high utilization is observed on a particular machine, the next job is routed to another machine. The results generated by reactive scheduling are often globally worse than those generated by globally oriented techniques – especially if there are only a few uncertainties in a production system. Reactive techniques were nevertheless dominant in production scheduling for many years (Baker 1974). The most important reason for this is that they do not require much computational power (Günther and Tempelmeier 2005). Another reason is their flexibility, which is coming to special importance when there is a significant amount of uncertainty in a production system.

The most common way to implement a technique based on the completely reactive approach is to use priority rules. This simply means to sort a given set of jobs or machines by certain parameters. Some popular priority rules are listed in Table 1 (see for example Panwalkar and Iskander (1977) for other rules).

Table 1: Often used priority rules

Abbreviation	Priority rule
FIFO	First In First Out
FOFO	First Off First On
SPT	Shortest Processing Time
LPT	Largest Processing Time
SRPT	Shortest Remaining Processing Time
LRPT	Largest Remaining Processing Time
EDD	Earliest Due Date

3.3 Predictive-Reactive Scheduling

The third type of techniques incorporating uncertainties can be described as predictive-reactive. As well as completely predictive methods these techniques build initial plans. But the resulting plans incorporate less uncertainty and are therefore more likely to turn out as inapplicable if parameters change. If this happens, the predictive-reactive approach changes the plan. To make these changes while the production process goes on, fast decision making is required. Thus, usually heuristics are used. The process of adapting an existing schedule to changing circumstances is often referred as rescheduling. For specific methods for predictive-reactive scheduling cf. Church and Uzsoy (1992), Alagöz and Azizoglu (2003) or Unal, Uzsoy, and Kiran (1997).

4 METHODOLOGY

4.1 Assumptions

Whether a predictive, a reactive or a predictive-reactive approach is preferable, differs from application scenario to application scenario. Thus, it is important to more precisely describe the production environment our approach should be applied to. The environment is considered to have the following characteristics:

- Flexible flow shop with more than one production stage and more than one machine at each of these stages
- Uniform machines at each production stage
- Highly flexible production environment with negligible set-up times
- High amount of uncertainty. An initial production plan has to be revised frequently during the production process
- Capital-intensive production of complex and usually high-priced products as they can be found in the semiconductor industry

4.2 Discussion of Scheduling Approaches

Using completely predictive techniques is not advantageous in the described production environment. The high amount of uncertainty would cause large buffers and hence significant inefficiency. This is of special importance if the buffered products lock up a large amount of capital. Incorporating all possible changes to the production parameters is just too expensive in the described environment.

Predictive-reactive approaches turned out to be beneficial in many different production environments. This however supposes a moderate amount of uncertainty. The more often unpredicted changes happen, the more often predictive-reactive techniques need to reschedule, and the less efficient they are. Lawrence and Sewell (1997) state predictive-reactive techniques are only advantageous over completely reactive methods, if the amount of uncertainty does not rise over a certain level. In the case of the described environment it can therefore be doubted that predictive-reactive approaches will turn out to be most efficient.

Completely reactive approaches have the ability to react quickly to frequent changes. They therefore appear to be suitable for the described environment with a high amount of uncertainty. A severe disadvantage of completely reactive approaches is however that they do not consider global results and therefore often produce unpredictable results of low quality. This problem is tackled in the residual paper.

4.3 Partly Reactive and Partly Predictive-Reactive Scheduling

Our approach does not fit well in the described classification of scheduling techniques as predictive, reactive and predictive-reactive. It should follow an approach which could be classified as partly reactive and partly predictive-reactive. Hence, the method should be able to immediately make decisions by considering local parameters only, and it should at the same time be globally oriented.

We made a central assumption: The described goal can be achieved by adapting the human process of decision making. Human planners would, if time is critical, not consider all parameters influencing a production process, but would partly decide by intuition. Humans learn by experience and therefore often know what is best in a situation even when there is no time for precisely calculating the outcome of the decision.

The system to be developed in this paper uses artificial intelligence in order to imitate the human process of decision making. It tries to gather as much experience as possible by a simulation process, done prior to the production process. During the running production it does not revert to a plan to make decisions, but uses the built knowledge base to classify the current situation and make the globally most promising decision. Thus, the system is on the one hand completely reactive (its decisions are only based on local and current parameters) and on the other hand predictive-reactive (its decisions are based on the process of knowledge gathering, which is done prior to the production process).

5 IMPLEMENTATION FRAMEWORK

5.1 Simulation System

To quickly and reliably gather information prior to the production process is of great importance for the success of the considered method. This is done by simulation. The simulation system used is d^{3}FACT insight, which has been developed by our institute since approximately seven years (see for example Dangelmaier et al. (2005); Dangelmaier et al. (2006) and Aufenanger et al. (2008)). The specific flexible flow shop for which a scheduling has to implemented is modeled in the simulation system. Then, many different – how many solely depends on the time being available prior to the production process – decisions are made randomly and their outcome is simulated. Subsequently, the decisions are classified as good (promising) or bad (non-promising) according to the global scheduling result.

5.2 Decisions to be Made

There are two kinds of decisions which have to be made during production in a flexible flow shop: First, when a job is ready to be processed at a certain stage, it has to be decided on which of the machines of the stage it should be processed (machine decision). And second, if there is more than one job ready to be processed at the stage, it has to be decided which of these jobs should be processed first (job decision).

To generalize the process of decision making, the technique which is to be developed should revert to the concept of priority rules. But, and that is an important difference to completely reactive scheduling in the traditional way, it should decide which priority rule is most likely to make the globally best decision in the current situation.

Because there are two kinds of decisions, we need to define two classes of priority rules from which the technique can choose. For the purpose of describing an example implementation framework we suggest to use the priority rules

- SPT: Shortest Processing Time first
- LPT: Longest Processing Time first

as possible priority rules for the job decision and

- Fastest: Fastest machine first
- Slowest: Slowest machine first

as possible priority rules for the machine decision. Since we present first results, we only use a few rules. The selection has to be enhanced in the future. The last decision is of special importance because of considering uniform machines, which work at different speeds. When considering identical machines, a machine can be selected randomly without harm. If the machines are unrelated, the priority rules as well as the process of generating conflict sets have to be adapted.

5.3 Building Conflict Sets

Building groups of jobs and machines from which the system to be developed is able to choose, is not trivial. In terms of job decision, a job which is currently ready to be processed does not only compete against other currently ready jobs for processing on the considered stage, but it also competes against jobs which would get ready to be processed during the time itself would be processed. In other words, it can be advantageous not to decide for processing a currently ready job, but to wait for another job to be ready.

Building conflict sets, out of which a priority rule (which is selected dynamically in our case) can choose, has been formalized by Giffler and Thompson (1960). As part of their algorithm to build all active schedules in a job shop environment they suggest to first select the job with the earliest possible finishing time C_{j^*} at the considered stage and then add to the conflict set all jobs with earliest possible start times $r_j < C_{j^*}$.

To decide if the heuristic presented by Giffler and Thompson (1960) can be used to build conflict sets in our case, it is important to answer two questions: Is the method applicable in flexible flow shop environments with their two kinds of decisions? And is it applicable in cases of reactive (and not only predictive) scheduling?

Giffler and Thompson (1960) partly answer the first question by themselves when stating: "Suppose that there are k machines available (...). Then (...) we can permit conflict sets to be as large as k without harm." This implicitly suggests to randomly allocate jobs to all available machines. Doing this does not cause any harm as long as the machines considered are identical. If they are uniform (doing the same work but at different speeds), this is not necessarily true. Therefore, machine allocation must not be done randomly.

Important hints on adapting the method of Giffler and Thompson (1960) to flexible flow shop scheduling are delivered by Nascimento (1993) and Chang and Sullivan (1990). They basically describe that good scheduling results are achievable in environments consisting of many flexible manufacturing systems by just considering all possible machine allocations in addition to job selection by the method of Giffler and Thompson (1960). Since flexible flow shops are a special kind of flexible manufacturing systems, this approach can be adapted to our case. Thus, job decision conflict sets are build by the method of Giffler and Thompson, and machine decision conflict sets just consist of all available machines at the considered stage. This leads to a process of building conflict sets as follows:

1. Build conflict set 1 consisting of all ready operations
2. Select from conflict set 1 the operation with the shortest processing time
3. Add to conflict set 1 all operations which are currently processed on the production stage preceding the considered one and will be ready for processing prior to the end of the previously chosen operation
4. Select an operation from conflict set 1 by a method to be defined
5. Build conflict set 2 consisting of all machines at the considered stage
6. Select a machine from conflict set 2 by a method to be defined
7. Start processing of the selected operation on the selected machine

In order to answer if the modified algorithm of Giffler and Thompson is applicable for reactive scheduling, we refer to the sequential nature of the original method the authors present. It builds schedules from start to end by only considering currently available information. Therefore, it is applicable for reactive scheduling.

5.4 Making Decisions

The most important part of the system to be developed in this paper is the process of making decisions. Its quality primarily affects the quality of the reactively developed scheduled. As already mentioned, the decisions to be made by an artificial intelligence method are reasonably abstract. It merely has to decide for a job and a machine selection priority rule. Specific jobs and machines are then chosen by the selected rule.

The method should make globally promising decisions on the basis of local information. Discovering which decisions are globally good in a given situation should be done by machine learning. Therefore, the outcome of as many different decisions as possible has to be simulated prior to the production process. This can for example be done by choosing job and machine selection priority rules randomly at every stage and then saving the decisions, descriptions of the situations they were made in, and the resulting global makespan. After many simulation runs of that kind it can be decided (by comparing the makespans) which simulation runs should be considered as good. The decisions made during these ”good” runs can then be regarded as examples of future decisions.

Discovering coherencies on the basis of a limited set of examples is an important sub-field of artificial intelligence and machine learning, which is referred to as classification. Classification is therefore used in the scheduling system developed in this paper when deciding which priority rules are most promising. There are many different classification methods. Selecting among them potentially has great impact on the scheduling results.

Among the most often used classification techniques are classification (decision) tree methods. These methods use example data to develop trees, in which branches represent certain attribute values. The branches lead to leaves, which represent decisions. An implementation is the supervised learning algorithm C4.5 (Quinlan 1993), which is an enhancement of older methods that build CARTs (classification and regression trees). In this paper we suggest to use the algorithm C4.5 for both necessary classification processes at each production stage: the one choosing a priority rule for the job decision and the one choosing a priority rule for the machine decision.

In order to classify a situation (in our case: find the most promising priority rules) each situation has to be described. In classification, describing a situation is done by referring to the values of its attributes. Therefore a set of attributes needs to be defined. In order to describe situations which can occur during the considered production process we suggest using the following attributes:

- The number of jobs in the currently considered job conflict set
- The standard deviation of processing times in the currently considered job conflict set
- The utilization of machines at the considered production stage

- The position of the considered production stage in the entire production process
- In the case of a job decision: the last made decision
- In the case of a machine decision: the job decision earlier made in the same situation

6 EVALUATION

6.1 Evaluation system

To evaluate our concept of an adaptive scheduling system for flexible flow shop production environments we developed an extension of the d^{3}FACT insight simulation system. It basically consists of a java class, which handles the routing (and therefore the described process of decision making) at every stage of the considered production environment. This class is called "FFS_knowledgeBased". It uses some other classes which are mentioned in figures 1 and 2.

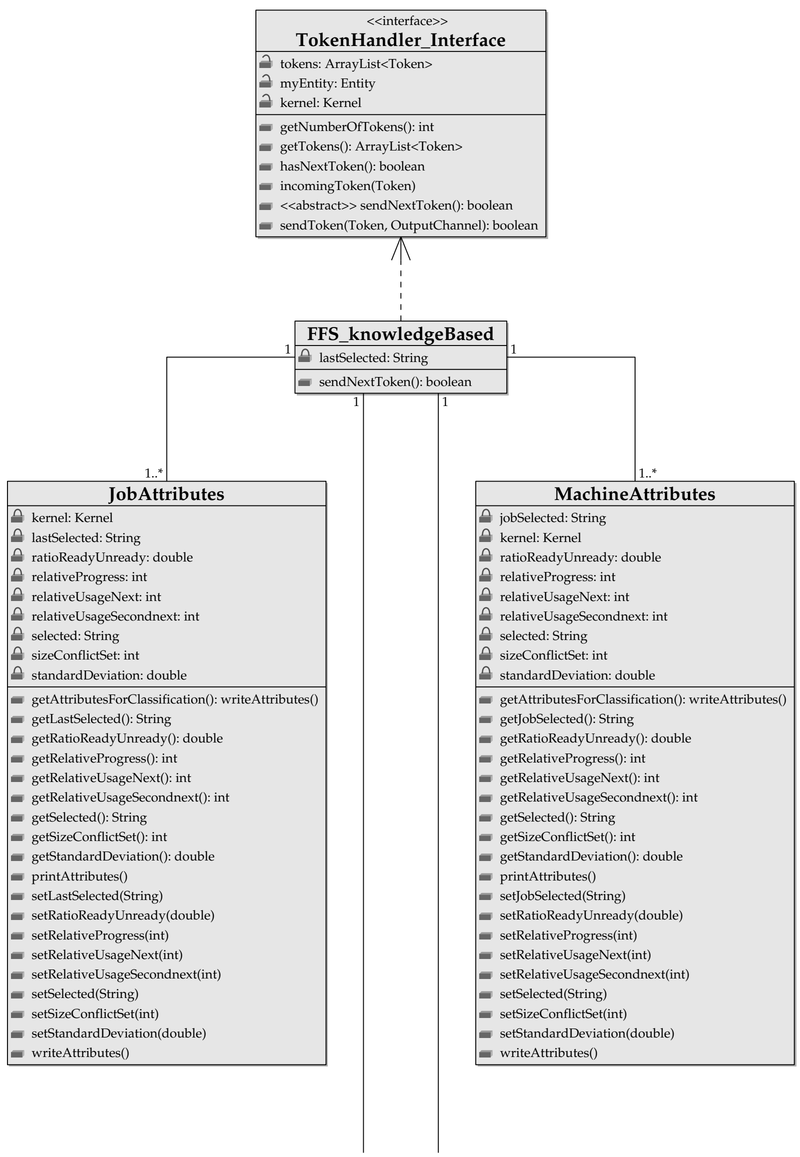

Figure 1: Classes of the evaluation system developed as an extension of d^{3}FACT insight. Part one.

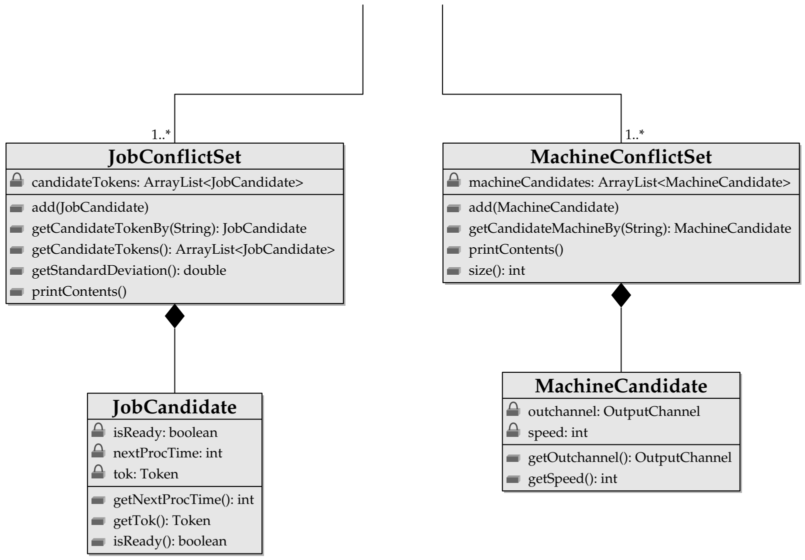

Figure 2: Classes of the evaluation system developed as an extension of d^3FACT insight. Part two.

6.2 Scheduling Results

The evaluation system was used to simulate the results of scheduling by the developed technique in different flexible flow shop production environments. These are primarily defined by their numbers of stages and machines at these stages. For each of the five different environments considered, we first used the evaluation system to gather training examples. Thus, we randomly generated a set of jobs and made the system select priority rules by chance during 100 simulation runs. After that, we selected the decisions of the 20 best simulation runs as examples of good decisions.

In a second step, we used the classifier to make the job and machine priority rule decisions on the basis of the gathered examples (knowledge). In order to obtain information about the adaptiveness of the developed technique, we did this with five variations of the flexible flow shop production environment used to gather training examples. The resulting makespans are listed in Table 2 and visualized in Figure 3. To be able to compare these results, we also list the makespans which have been achieved by using static priority rules and randomly chosen priority rules.

By analyzing the scheduling results, it can be asserted that the developed technique for adaptive scheduling in flexible flow shop production environments is in nearly all cases able to produce better results than static priority rules or scheduling by randomly choosing priority rules. Since there are many possible improvements, the basic approach seems promising. Positive impact on the scheduling results may for example be choosing other than the defined classification attributes, increasing the number of training runs or selecting another classification algorithm.

6.3 Calculation Time

It was our goal to develop a technique being suitable in situations in which production parameters change frequently. Hence, the developed technique has to make decisions without causing noticeable delays. This has been achieved. The time the process of classifying consumes is comparable to the time the application of a static priority rule would consume. Additional time is only needed prior to the production process. Since calculation time usually is not critical during that phase, this does not cause delays.

7 CONCLUSION

In this paper we described a technique which is able to adaptively schedule a production process in a flexible flow shop environment with uniform machines. The technique gathers information about the environment prior to the production process by simulation and using machine learning methods. It discovers which decision is most likely to cause a globally good result in a certain situation and uses this knowledge when it is used in real-world environments. Hence, it is able to make decisions exceptionally fast. It is used reactively without causing noticeable delays. For that reason it is adaptive while producing globally good results. It is a task of future research to enhance the technique by adding more sophisticated attributes for situation description or using other classifiers, for example a Naive Bayes classifier.

REFERENCES

Alagöz, O., and M. Azizoglu. 2003. Rescheduling of identical parallel machines under machine eligibility constraints. *European Journal of Operational Research* 149 (3): 523–532.

Aufenanger, M., W. Dangelmaier, C. Laroque, and N. Rüngener. 2008. Knowledge-based event control for flow-shops using simulation and rules. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 1952–1958. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Aytug, H., M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research* 161:86–110.

Baker, K. 1974. *Introduction to sequencing and scheduling*. New York: Wiley.

Blazewicz, J., K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. 2007. *Handbook of scheduling. from theory to applications*. Berlin, Heidelberg: Springer.

Brucker, P. 2007. *Scheduling algorithms*. 5. ed. Berlin: Springer.

Chang, Y.-L., and R. S. Sullivan. 1990. Schedule generation in a dynamic job shop. *International Journal of Production Research* 28 (1): 65–74.

Church, L. K., and R. Uzsoy. 1992. Analysis of periodic and event-driven rescheduling policies in dynamic shops. *International Journal of Computer Integrated Manufacturing* 5 (3): 153–163.

Dakin, R. 1965. A tree-search algorithm for mixed integer programming problems. *The Computer Journal* 8 (3): 250–255.

Dangelmaier, W., M. Aufenanger, C. Laroque, and D. Huber. 2005. d^{3}FACT insight - Ein Materialflusssimulator zum Erstellen und Simulieren komplexer Materialflussmodelle. In *18th Symposium Simulationstechnique*, ed. F. Hülsemann, M. Kowarschik, and U. Rüde, 530–535.

Dangelmaier, W., M. Aufenanger, C. Laroque, D. Huber, and K. Mahajan. 2006. A multi-user modeling environment to create and parametrize material flow models in d^{3}fact insight. In *Proceedings of the 13th European Concurrent Engineering Conference*, ed. U. Baake and E. Yucesan, 132–137.

Giffler, B., and G. L. Thompson. 1960. Algorithms for solving production scheduling problems. *Operations Research* 8 (4): 487–503.

Günther, H.-O., and H. Tempelmeier. 2005. *Produktion und Logistik*. 6. ed. Berlin: Springer.

Hopp, W. J., and M. L. Spearman. 2001. *Factory physics: foundations of manufacturing management*. 2. ed. Boston: Irwin/McGraw-Hill.

Johnson, S. M. 1954. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly* 1:61–68.

Kouvelis, P., R. Daniels, and G. Vairaktarakis. 2000. Robust scheduling of a two-machine flow shop with uncertain processing times. *IIE Transactions* 32 (5): 421–432.

Kyparisis, G. J., and C. Koulamas. 2006. Flexible flow shop scheduling with uniform parallel machines. *European Journal of Operational Research* 168 (3): 985–997.

Lawrence, S. R., and E. C. Sewell. 1997. Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. *Journal of Operations Management* 15 (1): 71–82.

Mehta, S., and R. Uzsoy. 1998. Predictable scheduling of a job shop subject to breakdowns. *IEEE Transactions on Robotics and Automation* 14 (3): 365–378.

Nascimento, M. A. 1993. Giffler and Thompson's algorithm for job shop scheduling is still good for flexible manufacturing systems. *The journal of the Operational Research Society* 44 (5): 521–524.

Panwalkar, S. S., and W. Iskander. 1977. A survey of scheduling rules. *Operations Research* 25 (1): 45–61.

Quinlan, J. R. 1993. *C4.5: programs for machine learning*. 1. ed. San Francisco: Morgan Kaufmann Publishers.

Sriskandarajah, C., and S. Sethi. 1989. Scheduling algorithms for flexible flowshops: Worst and average case performance. *European Journal of Operational Research* 43 (2): 143–160.

Unal, A., R. Uzsoy, and A. Kiran. 1997. Rescheduling on a single machine with part-type dependent setup times and deadlines. *Annals of Operations Research* 70:93–113.

Wang, H. 2005. Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. *Expert Systems* 22 (2): 78–85.

AUTHOR BIOGRAPHIES

MARK AUFENANGER studied Business Computing at the University of Paderborn, Germany. Since 2005, he is a research assistant at the group of Prof. Dangelmaier, Business Computing, esp. CIM. He is mainly interested in simulation of logistics systems, artificial intelligence and knowledge based systems. His email address is <Mark.Aufenanger@hni.upb.de>.

HENDRIK VARNHOLT studied Business Computing at the University of Paderborn, Germany. He is primarily interested in adaptive scheduling systems, real-time scheduling and artificial intelligence. His email address is <hevarnholt@gmx.net>.

WILHELM DANGELMAIER studied Mechanical Engineering at the University of Stuttgart, Germany. Since 1981, he was director and head of the Department for Corporate Planning and Control at Fraunhofer Institute for Manufacturing. In 1991, Dr. Dangelmaier became Professor for Business Computing at Heinz Nixdorf Institute, University of Paderborn, Germany. In 1996, he founded the Fraunhofer Center for Applied Logistics. His principal interests today are models and tools for distributed production systems. His email address is <Wilhelm.Dangelmaier@hni.upb.de>.

Table 2: The results of the evaluation process. Production environments are denoted by "M×N" with M representing the number of stages and N representing the number of machines at each stage. For each considered environment six variations were build. One of these was used for training. The others were used for application of the gathered knowledge. The results achieved when applying the knowledge were combined to mean values for each production environment. For comparison, results achieved when using static priority rules or randomly chosen priority rules are mentioned as well.

Problem	Job decision	Machine decision	Makespan in seconds (mean value)	Deviation from knowledge based result (mean value)
1x3	knowl. based	knowl. based	284.217	–
	SPT	fastest	295.417	4.05 %
	SPT	slowest	295.338	4.02 %
	LPT	fastest	284.217	0 %
	LPT	slowest	284.036	-0.07 %
	random	random	290.118	1.53 %
2x4	knowl. based	knowl. based	121.27	–
	SPT	fastest	174.054	46.13 %
	SPT	slowest	198.042	69.83 %
	LPT	fastest	123.962	2.32 %
	LPT	slowest	123.514	1.89 %
	random	random	127.355	5.17 %
3x2	knowl. based	knowl. based	715.714	–
	SPT	fastest	913.626	27.74 %
	SPT	slowest	931.557	30.26 %
	LPT	fastest	752.077	5.33 %
	LPT	slowest	749.999	5.19 %
	random	random	716.92	0.4 %
3x3	knowl. based	knowl. based	95.41	–
	SPT	fastest	100.47	5.8 %
	SPT	slowest	124.79	32.77 %
	LPT	fastest	104.5	10.04 %
	LPT	slowest	108.092	14.39 %
	random	random	105.115	10.61 %
4x4	knowl. based	knowl. based	111.702	–
	SPT	fastest	143.435	28.07 %
	SPT	slowest	219.575	97.2 %
	LPT	fastest	124.292	11.15 %
	LPT	slowest	142.842	29.51 %
	random	random	117.528	5.41 %

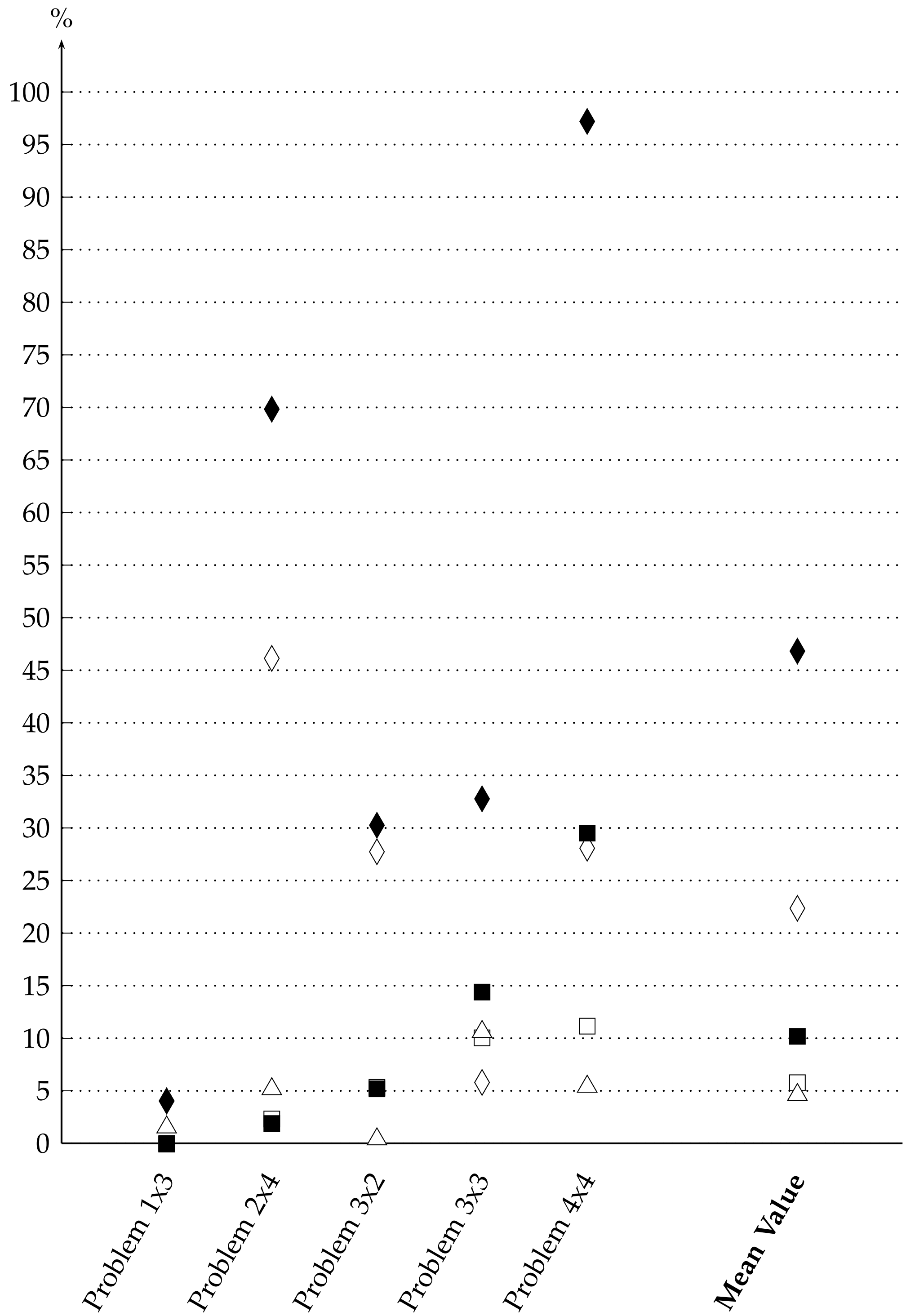

Figure 3: Visualization deviations (mean values) in percent from corresponding knowledge based results. ◊ denotes SPT/fastest. ♦ denotes SPT/slowest. □ denotes LPT/fastest. ■ denotes LPT/slowest. △ denotes randomly chosen priority rules.