

SIMULATION-BASED ADAPTION OF SCHEDULING KNOWLEDGE

Mark Aufenanger
Patrick van Lück

University of Paderborn
Business Computing, esp. CIM
Paderborn, 33098, GERMANY

ABSTRACT

Nowadays, markets are changing frequently and so are the orders that were placed. Therefore, the time from ordering a product until the delivery date becomes shorter and shorter. Furthermore, production systems are subject to different exogenous and endogenous disturbances like machine breakdowns, urgent orders, material failures and so on due to companies acting in a fast and complex world. Currently available scheduling and rescheduling mechanisms are lacking in solution quality or need large calculation times. Therefore, new self-adapting systems that are able to generate good solutions quickly and refine itself over time are needed. A new approach for a simulation based adaption mechanism for a knowledge based system is presented in this paper. Adaption of the knowledge based and the used classifier is supported by the mechanism. It is shown, that the solution quality increases when using the adaption mechanism instead of the native system without adaption component.

1 INTRODUCTION

Within a production system the schedule is the closest plan to the daily business (Blazewicz et al. 2007). Therefore a huge amount of time and computing power is spent before the start of the production process to simulate (Banks et al. 2000) and generate a mostly optimal schedule. When creating an optimal plan, no capacities of machines, workers and so forth will be wasted and the production system can work as effectively as possible, concerning the current status of orders or jobs. However, this assumption is only true, if the production situation is fixed for the time of the planned production process. Unfortunately, real life is not that predictable; disturbances, urgent jobs etc. happen all the time. It is seldom possible to create a new optimal scheduling plan within a running production system in time, because mostly all realistic sized scheduling problems are too complex (Hopp and Spearman 2007). The complexity of a problem is measured by the time it takes to solve it algorithmically (Pinedo 1994) and during the production process, time is the critical factor.

Following the stated situation, on the one hand it would consume much time to generate a rescheduling solution and because the production system has to be blocked during the calculation process the generation of products (income) is prevented within this calculation period. On the other hand, if solutions will be taken stepwise or quick and dirty by the use of fast heuristics, though time as the critical factor is eliminated, because of the low solution quality, money (income) is wasted. The throughput, cycle times etc. suffer because of the low rescheduling solution quality (Aytug et al. 2005).

For this reason, nowadays approaches in the research area of production planning have moved the focus to make planning more flexible. Production processes are no longer regarded as completely deterministic but as partially stochastic. It is realized, that very often a production schedule cannot be executed as planned. Consequently, researchers try to develop methods being able to quickly react on changes affect-

ing the production system while also being reliable. Such methods are described by Aytug et al. (2005). Among other methods the authors review techniques which utilize data (knowledge) gathered prior to the production process. By the use of this knowledge in combination with artificial intelligence algorithms the described methods try to predict the outcome of different decisions in a specific situation. However, the development of a knowledge based decision system to be applied as a flow control mechanism in a production environment is a step in the right direction, but the production system and its surroundings are refined constantly. Hence the control methods have to be refined during the use as well .

Therefore, new self-adapting decision support systems (Mönch and Pankratz 2008) are needed. They must be able to generate good solutions quickly and refine itself over time. A new approach for a simulation based adaption mechanism for a knowledge based re/scheduling system (Aufenanger et al. 2008) is presented in this paper. Adaption of the knowledge based and the used classifier is supported by use of this mechanism. It is shown, that the solution quality increases when using the adaption mechanism instead of the native system without adaption component. Furthermore, it outperforms several reference algorithms.

2 METHOD

As basic system framework (Aufenanger et al. 2008) the well known Giffler-Thompson algorithm was used (Giffler and Thompson 1960). Within this framework a Naïve-Bayes classifier is integrated to select jobs out of the conflict set. By doing so, a scheduling rule (here dispatching rules) for picking out the best fitting job in the current scheduling or rescheduling situation is selected by the use of the classification method. Figure 1 shows the system, where a classifier is used to generate the best fitting scheduling decision in each decision situation (point) within the problem scheduling tree. By doing so, instead of the evaluation of different alternative solutions paths, just one path through the tree that ends up in a good solution is worked out. Time for local searches and investigation of different solution trees is spared.

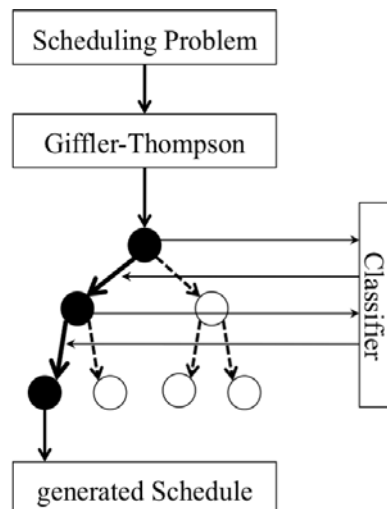


Figure 1: Guidance through the scheduling tree by the use of a classifier

The developed system can be used for schedule generation and rescheduling in job-, flow- and flexible-flow-shops. Depending on the target system, different specifications of the system internal components are used. For instance, in a flexible-flow-shop there is more than one classifier needed, because at every stage two decisions have to be taken: which job has to be selected and on which of the alternative machines the selected job should be machined. Furthermore, the system can differ from stage to stage, so normally the number of machines is different on each stage.

2.1 Knowledge Based Scheduling

The system for which the adaption method was developed consists of two main components (Aufenanger et al. 2008; Aufenanger, Varnholt, and Dangelmaier 2009): the learning (Training) and the scheduling component (adapted Giffler-Thompson). During the first phase, prior to the production process, the learning component is used. Here information on how to create good schedules is gathered and a knowledge base is created by inserting elements from good solutions relative to the target function. Good training solutions are created by the simulation of example problems with an optimal branch and bound algorithm (Brucker 2007; Brucker, Jurisch, and Sievers 1994). With this knowledge base a classifier is created. During the rescheduling phase the scheduling component is used. In each decision between two or more alternative jobs to schedule, the trained classifier decides which job is the best fitting one for the current situation, to achieve the target function (here makespan minimization) in the best way.

Figure 2 illustrates how the training of the system (classifier) works. A branch and bound method solves many scheduling problems as examples prior to the scheduling process. This takes a huge amount of time but time is not the critical factor compared to solution quality as long as the manufacturing system is not yet running. The developed system afterwards rebuilds the example solutions and every time there is more than one job to be scheduled at the same time at the same machine – the conflict set size is bigger than one – the job selected in the optimal example solution together with the current situation will be stored in the knowledge base. From this knowledge base a classifier can be created. Here the Naïve-Bays classifier was used because it is fitting best to our demands – quick classification, handles numerical attributes, etc. Nevertheless, another fitting classification method like a C4.5 tree can be used.

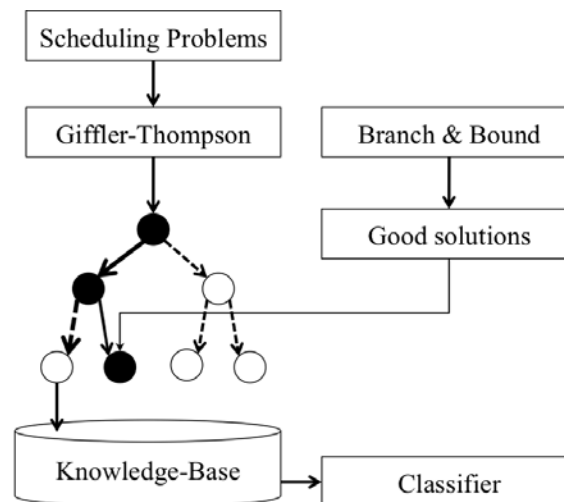


Figure 2: Setting up a scheduling knowledge base and classifier creation

2.2 Recognition of Situations

To store the situation based training information two demands have to be fulfilled. First the situations have to be recognized. This is done by the use of the following situation describing attributes:

- Relative size of the conflict set (Number of jobs within the conflict set / number of all jobs)
- Ratio of already scheduled working time to the total amount of working time of all jobs
- Ratio of already planned operations to the number of all operations
- Process ratio (percentage): Ratio of the second to the third attribute

- Machine load factor: Ratio of time, all machines have been worked to the time the machines had no jobs for working
- Machine load factor difference (percentage): Difference between the machine with highest and lowest utilization
- Minimal remaining time: Minimum for all unfinished jobs of the sum of remaining working times for the job
- Maximum lateness: Highest difference between simulation time and the time worked on a job
- Average lateness: Average of the lateness (as defined above) over all jobs
- Priority rule progress: Ratio between the working progress of the jobs selected by the priority rules

These attributes are used for job-shop systems. For flow-shop production systems the attributes are different. For instance the progress within the whole system and the stage size has to be monitored.

Secondly the selected job in a conflict situation can not be saved directly within the knowledge data-base because this way the example would not be generalizable enough. As described above we use priority rules for selecting the correct job instead of the direct job. Exemplarily the following rules are used within the system: SPT, LPT, MWR, LWR, LSO, and MSO. Certainly, other dispatching rules can be used as a selection rule within this system.

3 SIMULATION BASED KNOWLEDGE ADAPTION

For keeping the scheduling system effective over time it has to be adapted. Therefore, we developed a knowledge adaption for the application during the running production process. Figure 3 shows the interaction of the system components.

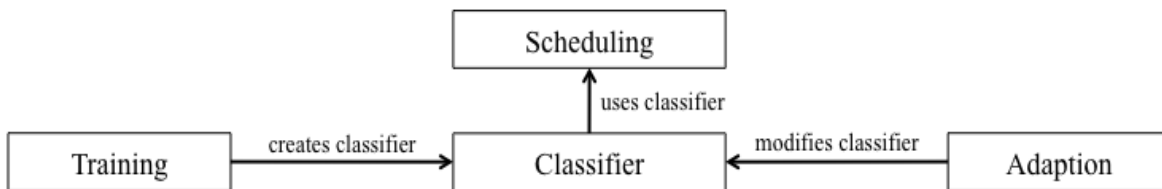


Figure 3: Interaction between adaption and classifier

While the classifier has been set up and trained during the initial phase, the adaption component adjusts the system by generating additional knowledge as well as testing different classifiers. Now the whole system consists out of three main components: Learning, scheduling, and adaption component.

3.1 Parallel Solutions Simulation

Generally, the basic method doesn't work chronologically when generating a schedule. By the use of a discrete event simulation the time reference can be created. Using additional information concerning the problem and gained through the simulative solution process, it can be determined on every point in time, on which planned steps modifications can be made. For the enhancement of the knowledge based scheduling system at each decision point an alternative solution was generated by a discrete event simulation system (Dangelmaier et al. 2005).

3.2 Knowledge Adaption

As soon as there are new good conflict set solution training examples, they will be added to the existing knowledge-base. This knowledge-base becomes especially adapted to the current problem and production

system. Every time a new plan is necessary, the knowledge-base will be used automatically. The so ad-justed data can also be used for the optimization of the not yet worked part of the schedule.

Figure 4 reflects the adaption process of adding additional training examples into the knowledge-base. By the use of this adaption method the solution quality of the generated schedules can be increased. By the knowledge-base adjustment to the current problem its deviation from problems to be solved can be avoided effectively. If the production system changes, e.g. by the replacement of slow machines, or the job characteristic changes over time, the knowledge-base is permanently effective and durable against this external changes, because of its constant adjustments.

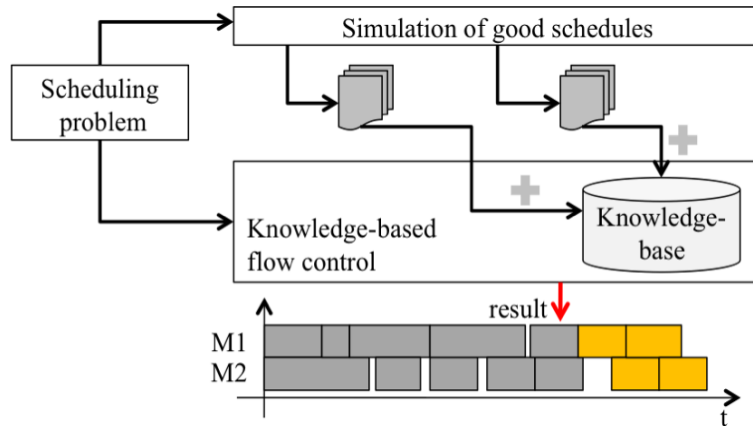


Figure 4: Knowledge-base adaption with additional scheduling examples

3.3 Classifier Selection

Figure 5 shows the algorithm for the adjustment of priority rules and so the classifier. Every time when the problem changes (new jobs arrive and so forth) or the knowledge-base was adapted, a new schedule is generated and the rule selection distribution is adjusted.

```

Data : SortedMap prules stores a mapping of solutions and corresponding priority
          rules in descending order of solution quality
while disruptionPoint reached do
  forall  $pr \in$  PriorityRuleSet do
    solution = solve problem using pr;
    prules.add(solution, pr);
  end
  for  $i = 0$  to  $prules.size()/2$  do
    pr = prules.get(i);
    increase number of alternative solutions for pr;
  end
  for  $i = prules.size()/2$  to  $prules.size()$  do
    pr = prules.get(i);
    decrease number of alternative solutions for pr;
  end
end

```

Figure 5: Adaption of priority rule set

During the schedule generation, different priority rules are used but only the best schedule is implemented. The adaption component saves every priority rule set's solution quality. On this basis the weighting, that defines the future use of priority rule sets for solution generation, is customized. Therefore, the number of alternative solutions will be increased for the more successful priority rule sets and will correspondingly be decreased for the rest of rule sets. Successful and not that successful are the one groups used in this context for simplicity. Therewith, priority rule sets, that are fitting better for the already scheduled part of the problem will be identified early and the calculation times will be adjusted as the most promising rule sets.

4 EVALUATION

The target function for the evaluation is the makespan minimization. During the evaluation of the developed adaption component, the following scheduling methods are used to create wide range of alternative solutions:

- Developed knowledge-based system with adaption component
- Branch and bound algorithm (Brucker, Jurisch, and Sievers 1994)
- Shifting bottleneck algorithm (Adams, Balas, and Zawack 1988)
- Native Giffler-Thompson method using the priority rules: SPT, LPT, LWR, MWR, LSO, and MSO
- Knowledge-based system without adaption component and using the priority rule sets: {LSO, MWR}, {SPT, LSO, MWR}, {LWR, MWR}, {LWR, MWR, LSO, MSO}, {SPT, LPT, LSO, MSO, LWR, MWR}, and {LSO, MSO}

We had evaluated the adaption method by applying it to different benchmark problems. Here exemplarily the results on the abz7, abz7-8, and ft10 problems are shown. The abz7 and abz8 problem are consisting out of 20 jobs and 15 machines (Adams, Balas, and Zawack 1988). For the abz7-8 problem we combined both problems (abz7 and abz8) by dynamically adding a new jobs out of the abz8 problem into the abz7 problem every 50 time units. The ft10 problem contains 10 jobs and 10 machines (Fisher and Thompson 1963).

By the use of the adaption method the solution quality could be increased in all test cases. As it can be seen in figure 6, the adaption methods leads to an absolute solution improvement between 1.3 and 3.3 percent compared to the systems solution quality without adaption method. This is a improvement of the relative solution quality of around 25 % for the ft10 and abz7 problem.

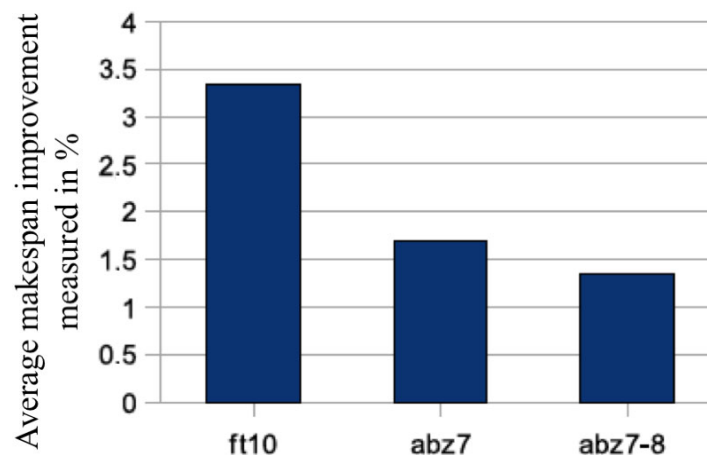


Figure 6: Schedule improvement by the use of the adaption method

Figure 7 illustrates the results for the abz7-8 benchmark problem. For this dynamic problem, the branch and bound algorithm is not able to create a solution. Therefore, the training was managed with the algorithmic solution of the static abz7 problem. For the comparison against the other scheduling methods the knowledge based solution with adaption component is the reference value now. As it can be seen, the native method without adaption mechanism performs the second best but did not reach the solution quality of the new method. PR-Set 1 and PR-Set 2 are representing the two best rule sets. The Giffler-Thompson heuristic without knowledge decision and fixed priority rule leads to the third best results (here also the two best rules are diagrammed).

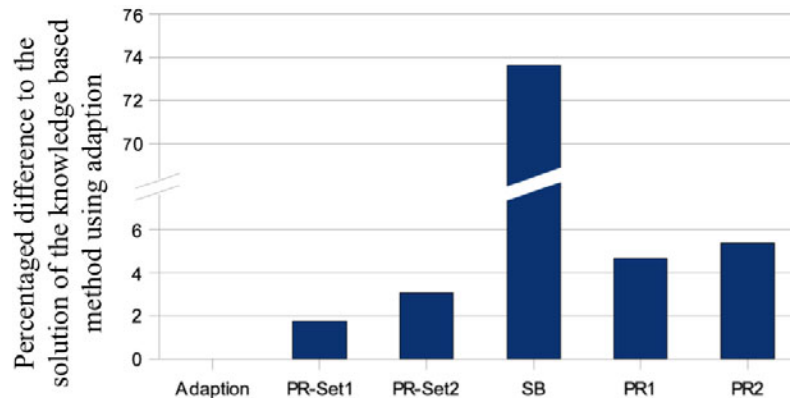


Figure 7: Comparison among different solution methods for the abz7-8

The worst results in comparison among all solutions were calculated by the shifting bottleneck heuristic. Its solution quality is up to 73 percent worse than the reference solution.

5 CONCLUSION

This paper handles the relevant topic of scheduling during the production process. An existing framework was enhanced by a knowledge base adaption method to permanently adapt the framework to the production system as well as the external environment. By the use of the adaption method, the gathered knowledge will be enhanced as well as the situational best fitting classifier will be selected. By the evaluation it could be shown, that the use of the adaption component leads to better scheduling solutions. The developed adaptive knowledge-based method generates very good solutions in a short amount of time, that exceeds the solution quality of the other reference scheduling heuristics.

Further work has to be done in the area of deleting bad training examples out of the knowledge-base, so that there is not just a increase of new good data, a decrease by the deletion of recognized bad old data is mandatory. Also, further research on situation describing attributes is necessary. They have to be created and evaluated automatically if the system should become accepted and is used in daily production life. Furthermore, more and different dispatching rules have to be tested on their applicability.

REFERENCES

- Adams, J., E. Balas, and D. Zawack. 1988. The shifting bottleneck procedure for job shop scheduling. *Management Science* 34(3): 391-401.
- Aufenanger, M., H. Varnholt, and W. Dangelmaier. 2009. Adaptive Flow Control in Flexible Flow Shop Systems - A Knowledge Based Approach. In *Proceedings of 2009 Winter Simulation Conference*, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 2164-2175. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

- Aufenanger, M., N. Lipka, B. Klöpper, and W. Dangelmaier. 2009. A Knowledge-Based Giffler-Thompson Heuristic for Rescheduling Job-Shops. In *2009 IEEE Symposium on Computational Intelligence in Scheduling*, 21-28.
- 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*, *European Journal of Operational Research* 161: 86-110.
- Aytug, H., S. Bhattacharyya, G. J. Koehler, and J. L. Snowdon. 1994. A review of machine learning in scheduling. *IEEE Transactions on Engineering Management* 41: 165-171.
- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2000. *Discrete-event system simulation*. 3rd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.
- 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*. 5th ed. Berlin: Springer
- Brucker, P., B. Jurisch, and B. Sievers. 1994. A branch and bound algorithm for the job-shop scheduling problem. *Discrete Appl. Math.* 49(1-3):107-127.
- Dangelmaier, W., M. Aufenanger, C. Laroque, and D. Huber. 2005. d3FACT insight -- Ein Materialflusssimulator zum Erstellen und Simulieren komplexer Materialflussmodelle. In *18th Symposium Simulationstechnique*, ed. F. Hülsemann, M. Kowarschik, and U. Rüde, 530-535. Erlangen: SCS Publishing House e.V.
- Fisher, M. L., and G. L. Thompson. 1963. Probabilistic learning combinations of local job-shop scheduling rules. In *Industrial Scheduling*, ed. J. Muth, and G. Thompson, 225-251. New York: Prentice-Hall.
- Giffler, B., and G. L. Thompson. 1960. Algorithms for solving production scheduling problems. *Operations Research* 8(4): 487-503.
- Hopp, W. J., and M. L. Spearman. 2007. *Factory Physics: Foundations of Manufacturing Management*. 3rd ed. Boston: Irwin/McGraw-Hill.
- Mönch, L., and G. Pankratz, ed. 2008. *Intelligente Systeme zur Entscheidungsunterstützung*. San Diego, Erlangen: SCS Publishing House.
- Pinedo, M. 1994. *Scheduling: Theory, Algorithms and Systems*. Prentice Hall College Div.

AUTHOR BIOGRAPHIES

MARK AUFENANGER is a Post-Doctoral researcher at the Heinz-Nixdorf-Institute at the Business Computing, esp. CIM, located at the University of Paderborn, Germany. He had studied Business Computing at the University of Paderborn. Thereafter, he worked as a researcher and gained PhD. Currently he works as a guest researcher at Product and Production development, Chalmers University of Technology, Sweden. His main research interests are in the area of discrete event simulation applied for the manufacturing industry, scheduling and rescheduling for production systems, and knowledge based systems. His email address is Mark.Aufenanger@hni.upb.de.

PATRICK VAN LÜCK studies computer science at the University of Paderborn, Germany. He has worked as a student assistant at the Chair of Business Computing, esp. CIM at the Heinz Nixdorf Institute located at the University of Paderborn, Germany. His main research interests are in the area of scheduling and rescheduling, knowledge based decisions, and discrete event simulation. His email address is van-lueck@gmx.net.