

# SIMULATION SUPPORT FOR RESCHEDULING

András Pfeiffer<sup>1</sup>, Botond Kádár<sup>1</sup>, László Monostori<sup>1,2</sup>

<sup>1</sup>Computer and Automation Research Institute Hungarian Academy of Sciences  
Kende u. 13-17, Budapest, H-1111, Hungary

<sup>2</sup>Department of Production Informatics, Management and Control, Faculty of Mechanical Engineering,  
Budapest University of Technology and Economics, Budapest, Hungary  
E-mail: pfeiffer@sztaki.hu

## KEYWORDS

Dynamic scheduling, rescheduling, simulation, stability

## ABSTRACT

The paper discusses the job shop scheduling problem and schedule measurement techniques, especially outlining the methods that can be applied in a dynamic environment. The authors propose a periodic rescheduling method by taking the rescheduling interval and schedule stability factor as input parameters into consideration. The proposed approach is tested on a simulated environment in order to determine the effect of stability parameters on the selected performance measures.

## INTRODUCTION

The broad goal of manufacturing operation management, such as a resource constrained scheduling problem, is to achieve a co-ordinated efficient behaviour of manufacturing in servicing production demands while responding to changes in shop-floors rapidly and in a cost effective manner.

In theory the aim is to minimize or maximize a performance measure. Regarding complexity, the job-shop scheduling problem (and therefore also its extensions), except for some strongly restricted special cases, is an NP-hard optimization problem (Baker 1998; Williamson et al. 1997).

The above mentioned job-shop scheduling is a static case, where all the information is available initially and it does not change over time. Most of the solutions in the literature concerning scheduling concentrate on this static problem. However, in many real systems, this scheduling problem is even more difficult because jobs arrive on a continuous basis, henceforth called dynamic job shop scheduling (DJSS). According to Rangsaritratamee, et al. (2004), previous research on DJSS using classic performance measures like makespan or tardiness concludes that it is highly desirable to construct a new schedule frequently so recently arrived jobs can be integrated into the schedule soon after they arrive.

Scheduling techniques addressing the dynamic – in the current case job shop – scheduling problem are called dynamic scheduling algorithms. These algorithms can be further classified as reactive and proactive

scheduling techniques. Depending on the environment, there may be deviations from the predictive schedule during the schedule execution due to unforeseen disruptions such as machine breakdowns, insufficient raw material, or difference in operator efficiency overriding the predictive schedule. The process of modifying the predictive schedule in the face of execution disruptions is referred to as reactive scheduling or rescheduling (Szelke and Monostori 1999). The reaction to the realised disruption generally takes the form of either modifying the existing predictive schedule, or generating a completely new schedule, which is followed until the next disruption occurs (Kempf et al. 2000).

The practical importance of the decision whether to reschedule or repair has been noted in (Szelke and Kerr 1994), while an additional categorization of scheduling techniques relating to the stochastic or deterministic characteristics of the problem can be found in (Kádár 2002).

It is important to outline, that while rescheduling will optimize efficiency using classic performance measures (makespan or tardiness) the impact of disruptions induced by moving jobs during a rescheduling event is mostly neglected. This impact is frequently called stability (Rangsaritratamee et al. 2004; Cowling and Johansson 2002). In related previous works, the number of times rescheduling takes place was used by Church and Uzsoy (1992) as the measure of stability and it was suggested that a more frequent rescheduling means a less stable schedule. Other approaches defined stability in terms of the deviation of job starting times between the original and revised schedule and the difference of job sequences between the original and revised schedules. One of the shortcomings of these approaches is that they ignore the fact that the impact of changes increases as they are made closer to the current time. Rangsaritratamee, et al. (2004) propose a method which addresses DJSS based on a bicriteria objective function that simultaneously considers efficiency and stability, and so let the decision maker to strike a compromise between improved efficiency and stability. In the approach, two dimensions of stability are modelled. The first captures the deviation of job starting times between two successive schedules and the second reflects how close to the current time changes are made.

Vieira, et al. (2000) presents new analytical models that can predict the performance of rescheduling strategies and quantify the trade-offs between different performance measures. Three rescheduling strategies are studied in a parallel machine system: periodic, event-driven and hybrid, similarly to the work (Church and Uzsoy 1992). They realized that there is a conflict between avoiding setups and reducing flow time, and the rescheduling period affects both objectives significantly, which statement is coincident concluded by Rangsaritratamee, et al. (2004).

In order to decide what action we should take in response to an event, we should have some idea of the value of our current schedule. In the following section evaluation classes of production schedules are introduced.

### **EVALUATION OF PRODUCTION SCHEDULES**

This part of the paper discusses the problem of schedule measurement and especially outlines the techniques which can be applied in a dynamic environment.

The quality of factory scheduling, generally, has a profound effect on the overall factory performance. As stated in (Kempf et al. 2000), an important aspect of the schedule measurement problem is whether an individual schedule or a group of schedules is evaluated. Individual schedules are evaluated to measure its individual performance. For a predictive schedule, the result may determine whether it will be implemented or not. There might be different reasons for evaluating a group of schedules. One of them is to compare the performance of the algorithms with which the different schedules were calculated. The comparison of different schedule instances against different performance measures is an other option in the evaluation of a set of schedules for the same problem.

According to Kempf, et al. (2000), relative comparison assumes that for the same initial factory state two or more schedules are available, and the task is to decide which is better. The task is to decide which one is better from two schedules, or which one is the best from a group of schedules generates additional questions. In a complex manufacturing environment different schedules will probably perform better against different performance measures. Therefore, the selection of the best schedule will always depend on the selected performance measure(s) and thus, on the external constraints posed by the management of the enterprise.

An absolute measurement of schedule quality consists in taking a particular schedule on its own and deciding how „good” it is. This requires some set of criteria or benchmarks against which to measure.

Regarding the predictive schedules, a set of decisions is made on the base of estimates on future events, without knowing the actual realizations of the events in question until they actually occur. Taking this fact into

consideration, Kempf, et al. (2000) differentiate between the static and dynamic measurements of predictive schedules. A static measurement means the evaluation of the schedule independently of the execution environment.

Contrary to static measurement, the dynamic measurement of a predictive schedule is more difficult. In this case, beyond the static quality of the schedule, the robustness of the schedule against uncertainties in the system should also be taken into consideration.

Another aspect in the evaluation of schedules is the state of the manufacturing system after the execution of the schedule. In (Kempf et al. 2000) these parameters are compared as state measurements, which evaluate the end effects of the schedule at the end of the schedule horizon.

Regarding the evaluation classes listed above, a dynamic measurement of individual predictive schedules will be presented in the following sections.

### **SIMULATION IN DYNAMIC SCHEDULING**

Simulation captures those relevant aspects of the production planning and scheduling (PPS) problem, which cannot be represented in a deterministic, constraint-based optimization model. The most important issues in this respect are uncertain availability of resource, uncertain processing times, uncertain quality of raw material, and insertion of conditional operations into the technological routings.

The features provided by the new generation of simulation software facilitate the integration of these tools with the production planning and scheduling systems. Additionally, if the simulation system is combined with the production database of the enterprise it is possible to instantly update the parameters in the model and use the simulation parallel to the real manufacturing system supporting and/or reinforcing the decisions on the shop-floor.

The reason of the intention to connect the scheduler to a discrete event simulator was twofold. On the one hand, it serves as a benchmarking system to evaluate the schedules on a richer model; on the other hand, it covers the non-deterministic character of the real-life production environment. Additionally, in the planning phase it is expected that the statistical analysis of schedules should help to improve the execution and support the scheduler during the calculation of further schedules

In the proposed architecture the simulation model replaces a real production environment, including both the manufacturing execution system and the model of the real factory.

Simulation also generates continuously new orders into the system, while these new orders are scheduled and released by the scheduler.

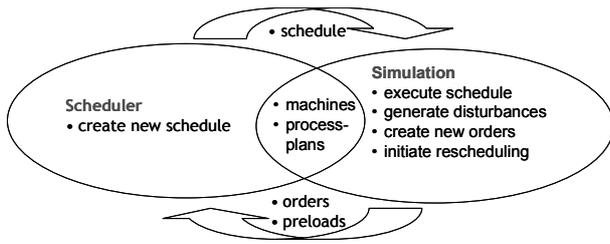


Figure 1. The rescheduling process initiated from the simulation side

The outline of the developed architecture is presented in Figure 1. Rescheduling action can be initiated when an unexpected event occurs or if a main performance measure bypasses a permissible threshold.

The dynamics of the prototype problem have been constructed to preserve realism as closely as possible and make the problem manageable for analysis.

This way simulation is capable for interaction with a specified scheduler, because all the required parameters are available any time for both systems, and so formulating an environment for further analysis on e.g. order pattern or sensitivity on significant parameters.

## PROPOSED METHOD

The study analysis the impact of the rescheduling interval and the rate of schedule modification on classical performance measures as system load, efficiency as well as stability in a single machine prototype system.

From the practical point of view in scheduling it is not possible to create schedules in every minute, however, the (theoretically) best performance of the whole system could be realized if schedule could be able to adapt to any changes, disruptions occurring in real-time. Most industrial planning and scheduling systems create schedules at idle time of the production e.g. at nights, while creating schedules for larger job-shop mostly requires a lot of computational time.

The process of modifying the schedule in the face of execution disruptions is referred to as reactive scheduling or rescheduling – detailed in the previous sections. The reaction to the realized disruption generally takes the form of either modifying the existing (predictive) schedule, or generating a completely new schedule, which is followed until the next disruption or rescheduling event occurs. The first technique is described in (Vieira et al. 2000; Rangsaritratamee et al. 2004), while the second is presented in (Bidot et al. 2003; Cowling and Johansson 2002). The importance of stability is outlined in the selected studies (see “monotonic and non-monotonic approach” in (Bidot et al. 2003), or “2D stability” in (Rangsaritratamee et al. 2004). The most important point is that while scheduling will optimize the efficiency measure, the strategy generates schedules that are often radically different from the previous ones. From the practical point of view the scheduling technique mentioned first seems to be better, while in industrial applications

constructing completely new schedules during schedule execution process must be avoided.

Schedule modification can be executed in given time periods (periodic rescheduling strategy), or related to specified events occurring during schedule execution (event-driven rescheduling strategy). Combining the two methods hybrid rescheduling strategy can be defined under which rescheduling occurs not only periodically but also whenever a disturbance is realized in the system (e.g. machine failures, urgent orders).

Define the time at which a new schedule is constructed as the rescheduling point and the time between two consecutive rescheduling points as the rescheduling interval (RI). At each rescheduling point, all jobs from the previous schedule that remained unprocessed are combined with the jobs that arrived since the previous rescheduling point and a new schedule is built.

In the previous sections the problem of DJSS and related stability measurements were introduced as well as the proposed simulation environment for dynamic rescheduling. In order to prove whether the rescheduling interval and the newly introduced variable schedule stability factor have a significant effect on schedule quality as well as stability an experiment for simulated single machine case was realized.

## Efficiency

The system to be scheduled is a single machine system with continuous job arrivals, but without any due date limitations. According to Baker (1974), the current scheduling problem can be classified as a single machine sequencing case with independent jobs and without due dates. In these situations the time spent by a job in the system can be defined as its flow time and the “rapid turnaround” as the main scheduling objective can be interpreted as minimizing mean flow time. The objective function is calculated as follows:

$$\bar{F} = \frac{1}{n} \sum_{j=1}^n (c_j - r_j) \quad (1)$$

where

$\bar{F}$  is the mean flow time

$n$  is the number of total arrivals

$r_j$  is the point in time at job  $j$  entered the system

$c_j$  is the completion time of job  $j$ , calculated when job  $j$  leaves the system

## Stability

In our study stability is calculated for each available job in the system during schedule calculation by giving penalty (PN) values, using the relation  $penalty = starting\ time\ deviation + actuality\ penalty$ . Starting time deviation is the difference between the start time of the job at the new and previous rescheduling points. Actuality penalty is related to a penalty function associated with deviation of the start time of the job from the current time. Penalty values are only calculated in case starting time deviation is greater than 0. A schedule with less penalty value can be considered as a

more stable schedule. The mean value of stability  $\overline{PN}$  is calculated for all schedules as follows:

$$\overline{PN} = \frac{1}{n_{pn}} \sum_{j \in B} \left[ |t'_j - t_j| + \frac{100}{\sqrt{t_j - T}} \right] \quad (2)$$

where

$B$  is the set of available jobs  $j$  that have not begun processing yet and  $|t'_j - t_j| > 0$   
 $n_{pn}$  is the number of the elements in  $B$   
 $t_j$  is the estimated start time of job  $j$  in the current schedule  
 $t'_j$  is the estimated start time of job  $j$  in the successive schedule  
 $T$  is the current time

### Schedule Stability Factor

When minimizing the objective function Equation (1), in a single machine case the optimal dispatching rule to be selected is SPT (shortest processing time) detailed in (Baker 1974). In the current case we use a truncated shortest processing time (TSPT) rule, in which the schedule stability factor (SF) can be introduced as the measure of the importance of schedule continuity or monotony. SF is the continuity rate of the schedule creation. In case SF equals zero, the new schedule may completely differ from the previous one, in case SF equals 1 the “old” jobs in the successive schedule must have the same position as in the previous one.

### Schedule Creation

SPT based scheduling means, that the priorities of the available activities are calculated by taking only the length of the processing time into consideration. On the other hand, the TSPT rule we introduce – see Equation (3) – generates schedules using SF in order to override the priorities of the activities given by the SPT rule, this way ensuring a more stable schedule. Each priority must have an integer value and it is calculated as follows:

$$prio'_j = (prio_j \times SF + prio_{j,SPT} \times (1 - SF))_{INT} \quad (3)$$

where

$A$  is the set of available jobs  $j$  that remained unprocessed in the previous schedule  
 $prio'_j$  is the modified priority of job  $j$  ( $j \in A$ ) in the successive schedule  
 $prio_j$  is the priority of job  $j$  in the previous schedule  
 $prio_{j,SPT}$  is the temporary priority of job  $j$  calculated using SPT rule

At each rescheduling point the following scheduling procedure is executed:

1. new jobs are added to set  $A$
2. create a priority list of jobs in set  $A$  by using SPT rule

3. compare current and previous priorities for “old” jobs and calculate new priorities by using Equation (3)
4. add remaining priorities to new jobs and sort the list by priority, calculate penalties by using Equation (2)
5. apply successive schedule and continue the schedule execution until the next rescheduling point defined by RI, then return to 1.

## ANALYSIS AND EXPERIMENTAL RESULTS

The above mentioned method was tested on a simulated single machine prototype system in order to measure the characteristics of stability measures in a simple environment.

The simulation system was developed using eM-Plant object oriented, discrete event driven simulation tool, which will be helpful during the extension of the current problem to larger, job shop problems.

In single machine case, minimizing mean flow time we applied SF and RI as input at given shop utilization levels. As output we considered  $\overline{F}$ ,  $n_{pn}$  and total penalty which is the sum of all  $\overline{PN}$  values multiplied by  $n_{pn}$  calculated at the end of each simulation run.

It was experimentally determined that the results from the first 2000 arrivals should be eliminated from computations to remove transient effects. Hence, each simulation run in this study consisted of 12000 arrivals of which the final 10000 were used to compute the performance and stability measurements reported. Each experiment was replicated 10 times to facilitate statistical analysis.

The interarrival time ( $b$ ), i.e. the average time between arrivals for jobs and are generated from exponential distribution with mean calculated using Equation (4):

$$b = \frac{\overline{p} \times n_o}{U \times m} \quad (4)$$

where

$\overline{p}$  is the mean processing time per operation  
 $n_o$  is the number of operations in a job, in the current case equals 1  
 $U$  is shop utilization level  
 $m$  is the number of machines in the system, in the current case equals 1

### Experiment 1

The main goal of Experiment 1 was to analyse the impact of system utilization level on  $\overline{F}$ , where SF was set to 0. Figure 2 shows, that both the system utilization and RI have a significant effect on  $\overline{F}$ . In the following experiment, where stability is examined we would like to use a relatively high utilization level in order to provide as much work-in-process as possible.

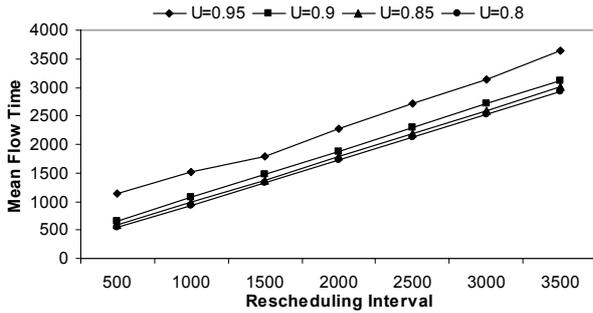


Figure 2. Effect of rescheduling interval and utilization level on mean flow time

As it is expected, extremely high utilization level lead to undesirable system instability, namely increasing the standard deviation of the resulted values and worsening the quality of the experimental results. The maximum acceptable value for  $U$  in the current case is 0.9.

### Experiment 2

The aim of Experiment 2 was to prove the assumption that applying the proposed stability criterion increases the stability of schedule execution however it reduces schedule efficiency. As a second scope of the experiment the effect of schedule stability factor on performance measurements was analysed.

In this experiment  $U = 0.9$  and  $p = 140$  with a triangle distribution  $\{140, 1, 300\}$ , then the mean of  $b$  equals 160. Three rescheduling interval were considered 500, 2000 and 3500 to have results from a wide range of RI. The second group of input parameters was SF, set to 0, 0.25, 0.5, 0.75 and 1.

As we assumed, the lengthening of the rescheduling interval increases stability but decreases the efficiency of the system. Figure 3 shows the illustrative results where SF was set to 0. Efficiency measurement  $\bar{F}$  is represented by the linear increasing dotted line, while the penalty values of the stability measurement are represented by the continuous line having a negative steepness. The penalty values decreased, because a higher number of modification made in the schedule at RI=500 with lower  $\overline{PN}$  values resulted a greater product than the same parameters at RI=3500.

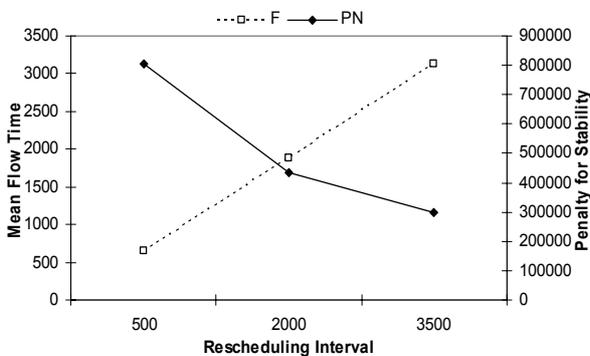


Figure 3. Effect of rescheduling interval on mean flow time and penalty values, in case SF=0

The effect of the parameter SF on penalty values given for stability and efficiency measurement  $\bar{F}$  at different rescheduling intervals are shown in Figure 4 and Figure 5.

Figure 4 shows for all RI curves, that the values increase in a monotonic way, i.e. increasing SF decreases system performance (increasing  $\bar{F}$ ) in each case. Comparing the results to SF=0, in case SF was set to 1, the outcome of the simulation showed an 8% increase of the performance measurement  $\bar{F}$ , in case RI was set to 500. Analyzing the other two cases, when RI was equal to 2000 and 3500, the performance of the system worsened only a few percent. Using these results it can be stated, that the negative effect of a higher SF level on  $\bar{F}$  decreases as the length of the rescheduling interval is growing.

On the other hand, penalty values decreased significantly at each RI (see Figure 5), because the higher SF values reduced the total  $\overline{PN}$  values, i.e. enabled less modification in the schedule.

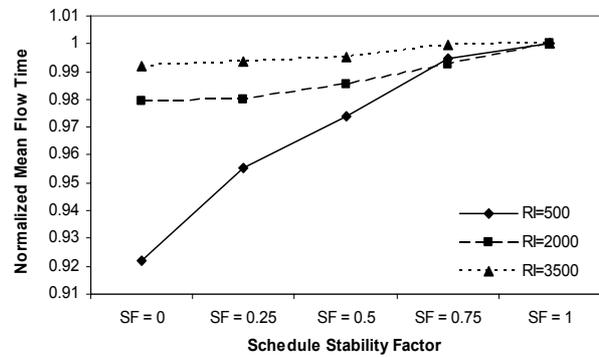


Figure 4. Effect of SF on normalized mean flow time at different rescheduling intervals

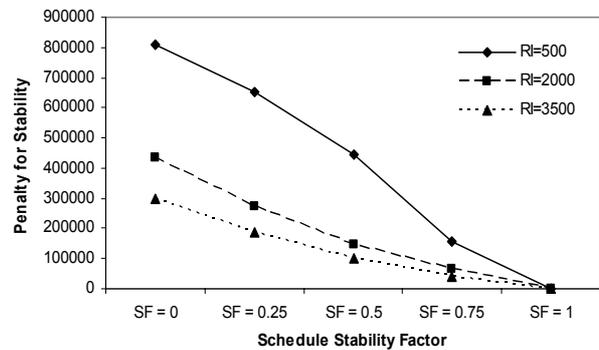


Figure 5. Effect of SF on penalty values at different rescheduling intervals

Comparing  $\overline{PN}$  values at different SF and RI parameter settings, it is interesting, that a penalty value given for SF=0 and RI=3500 is less than a penalty value for SF=0.5 and RI=500, while the efficiency is much better for RI=500.

Applying a limit for penalty values, e.g. let total  $\overline{PN}$  be  $ab \cdot 2 \cdot 10^6$ , then the optimal SF values can be selected for the given rescheduling intervals RI=500, 2000 and 3500. These values from Figure 5 are 0.7, 0.4 and 0.25 respectively.

## CONCLUSIONS

The paper discussed the job shop scheduling problem and schedule measurement techniques, especially outlining the methods that can be applied in a dynamic environment. The results of the simulation study based on the proposed architecture showed that both rescheduling interval and the newly introduced variable schedule stability factor have a significant effect on schedule quality as well as stability. In case applying limitations for stability, then for the given rescheduling intervals the optimal SF values can be determined. This significantly improves stability measurements but inconsiderably reduces system performance.

## FUTURE WORK

We would like to extend this experiment to a multi machine job shop system, using the results on stability gathered in this study. We propose a hybrid rescheduling strategy in a dynamic job shop environment defining two types of rescheduling events. The first type is done periodically (e.g. daily or weekly) using RI, releases new orders and involves tasks associated with order release. The second type is done when a disturbance occurs. It does not release new orders but instead reassigns work to off-load a down machine or utilize a newly-available one.

We assume that finding the appropriate schedule stability factor for each given rescheduling situation may result a compromise between the stable schedule execution and schedule quality.

## REFERENCES

- Baker, K. R. 1974. Introduction to sequencing and scheduling, John Wiley & Sons, USA.
- Baker, A. D. 1998. "A Survey of Factory Control Algorithms That Can Be Implemented in a Multi-Agent Hierarchy: Dispatching, Scheduling, and Pull". *Journal of Manufacturing Systems*, Vol. 17, 297-320.
- Bidot, J., P. Laborie, J. C. Beck, T. Vidal. 2003. "Using simulation for execution monitoring and on-line rescheduling with uncertain durations". *Proceedings of the ICAPS'03 Workshop on Plan Execution, Trento, Italy*.
- Church, L. K.; R. Uzsoy. 1992. "Analysis of periodic and event-driven rescheduling policies in dynamic shops". *International Journal of Computer Integrated Manufacturing*, Vol 5(3), 153-163.
- Cowling, P.; M. Johansson. 2002. "Using real time information for effective dynamic scheduling". *European Journal of Operational Research*, Vol 139, 230-244.
- Kádár, B. 2002. Intelligent approaches to manage changes and disturbances in manufacturing systems. PhD thesis. Technical University of Budapest, Hungary.
- Kádár, B.; A. Pfeiffer; L. Monostori. 2004. "Discrete event simulation for supporting production planning and scheduling decisions in digital factories". *Proceedings of the 37th CIRP International Seminar on Manufacturing Systems; Digital enterprises, production networks*, Budapest, Hungary, 444-448.
- Kempf, K.; R. Uzsoy; S. Smith; K. Gary. 2000. "Evaluation and comparison of production schedules". *Computers in Industry*, Vol 42, 203-220.
- Law, A.; D. Kelton. 2000. Simulation modelling and analysis, McGraw-Hill, 669-672.
- Rangaritratsamee R.; W. G. Ferrell Jr.; M. B. Kurz. 2004. "Dynamic rescheduling that simultaneously considers efficiency and stability". *Computers & Industrial Engineering*, Vol 46(1), 1-15.
- Szelke, E.; R.M. Kerr. 1994. „Knowledge based reactive scheduling state-of-the-art”. *Int. Journal of Production Planning and Control*, Vol 5 (March-April), 124–145.
- Szelke, E.; L. Monostori. 1999, Modeling Manufacturing Systems, Chap., Reactive scheduling in real-time production control. Springer, Berlin, Heidelberg, New York, 65-113.
- Vieira G. E., J. W. Herrmann, E. Lin. 2000. „Predicting the performance of rescheduling strategies for parallel machines systems”. *Journal of Manufacturing Systems*, Vol 19(4), 256-266.
- Williamson, D. P.; L.A Hall; J.A. Hoogetveen; C.A. Hurkens; J.K. Lenstra; S.V. Sevastjanov; D.B. Shmoys. 1997. "Short Shop Schedules". *Operations Research*, Vol. 45, 288-294.

## AUTHOR BIOGRAPHIES

**ANDRÁS PFEIFFER** received his M.Sc. degree in mechanical engineering in 2002, from the Technical University of Budapest. Currently he is a Ph.D. student at the Intelligent Manufacturing and Business Processes Research Group in the Computer and Automation Research Institute Hungarian Academy of Sciences (SZTAKI). His current interest includes production planning and scheduling, simulation and emulation of logistic systems. E-mail: pfeiffer@sztaki.hu, webpage: <http://www.sztaki.hu/~pfeiffer>.

**BOTOND KÁDÁR** is senior research fellow at Intelligent Manufacturing and Business Processes Laboratory of Computer and Automation Research Institute, Hungarian Academy of Sciences (SZTAKI). He obtained his M.Eng. and Ph.D degrees from the Technical University of Budapest, Hungary in 1993 and 2002 respectively. Since obtaining his PhD, he has been involved in research and development projects and education at Technical University of Budapest. His current interest includes production control, simulation and multi-agent approaches for production engineering and manufacturing systems. Dr. Botond Kádár is author or co-author of 55 publications. E-mail: kadar@sztaki.hu, webpage: <http://www.sztaki.hu/~kadar>.

**LÁSZLÓ MONOSTORI** has been with the Computer and Automation Institute of the Hungarian Academy of Sciences (SZTAKI) since 1977, now he serves as Deputy Director Research. He is also the head of the Department on Production Informatics, Management and Control at the Budapest University of Technology and Economics. He is an Active Member of the International Institutions for Production Engineering Research (CIRP) and Chairman of the Scientific-Technical Committee on Optimisation of Manufacturing Systems, Vice President of the International Society of Applied Intelligence (ISAI), Chairman of the Technical Committee on Manufacturing Modelling, Management & Control; International Federation of Automatic Control (IFAC). He is associate editor of Computers in Industry, CIRP Journal of Manufacturing Systems, IEEE Transactions on Automation Sciences and Engineering (T-ASE), and member of the editorial boards of some prestigious international scientific periodicals, such as the CIRP Annals. E-mail: laszlo.monostori@sztaki.hu, webpage: <http://www.sztaki.hu/~monostori>.