

ADAPTIVE SCHEDULING THROUGH MACHINE LEARNING-BASED PROCESS PARAMETER PREDICTION

M. Frye¹, D. Gyulai^{2*}, J. Bergmann², R. H. Schmitt³

¹Fraunhofer Institute for Production Technology (IPT), Aachen, Germany

²Institute for Computer Science and Control (SZTAKI), Hungarian Academy of Sciences (MTA), Budapest, Hungary

³Laboratory for Machine Tools WZL RWTH Aachen University, Aachen, Germany

*Corresponding author; e-mail: david.gyulai@sztaki.mta.hu

Abstract

Detailed manufacturing process data and sensor signals are typically disregarded in production scheduling. However, they have strong relations since a longer processing time triggers a change in schedule. Although promising approaches already exist for mapping the influence of manufacturing processes on production scheduling, the variability of the production environment, including changing process conditions, technological parameters and the status of current orders, is usually ignored. For this reason, this paper presents a novel, data-driven approach that adaptively refines the production schedule by applying Machine Learning (ML)-models during the manufacturing process in order to predict the process-dependent parameters that influence the schedule. With the proper prediction of these parameters based on the process conditions, the production schedule is proactively adjusted to changing conditions not only to ensure the sufficient product quality but also to reduce the negative effects and losses that delayed rescheduling would cause. The proposed approach aims on minimizing the overall lateness by utilizing an active data exchange between the scheduling system and the predictive ML-models on the process level. The efficiency of the solution is demonstrated by a realistic case study using discrete event simulation.

Keywords:

Artificial intelligence; Machine learning; Data analytics; Adaptive scheduling; Process parameter prediction; Process optimization; Job shop scheduling

1 INTRODUCTION

Industrial efforts and the progress in digitalization foster the application of data-driven modelling (DDM) in the production environment. DDM offers the potential to optimize processes up to complete process chains [Stein 2018]. Although the number of projects in the area of digitalization has increased, many companies still hesitate to boost DDM. This is due to limited resource opportunities or a lack of confidence of the management whether DDM will deliver the expected results [Schröder 2015]. For typical management objectives, the expected results are associated with increasing productivity, reducing costs and increasing service levels [Bughin 2017]. Therefore, DDM has to contribute achieving these management goals.

The paper presents a proof-of-concept study of cost-reduction and service-level-increase to be supported by DDM approaches applied in production scheduling. A new adaptive scheduling method is presented, which strongly relies on the prediction of remaining processing times. The main idea of the procedure is based on the combination of ML-model predictions and event-driven, adaptive rescheduling. As a first step of the presented approach, process parameters are acquired and monitored through sensor technology. As an assumption, the applied sensor technology is capable of providing descriptive data of the

process in real time. After acquisition, data is preprocessed and prepared. Then, data is fed into ML-models that provide the opportunity of estimating the remaining processing times. Although, accurate prediction can be solely meaningful, its real impact regarding the business value relies in the elevation of process information to a higher level in the decision-making hierarchy. This enables to optimize the entire process chain, which in this case is provided in the form of a robust production schedule.

The goal is to reduce idle times and waste by adaptively optimizing the production schedule at the earliest possible time when any deviation from the expected process completion is identified by the ML-models. If the entire order set is rescheduled in the beginning, the new schedule will deliver less total waste than implementing the rescheduling after the realization of the total lateness at the end of the given process.

The presented proof-of-concept relies on a discrete-event simulation (DES)-model of a realistic production system, combined with a scheduling model in a common application. Through a set of scenarios, the viability of applying ML-based process parameter prediction in scheduling is proven, namely that the proposed method can reduce the overall waste, compared to traditional rescheduling strategies.

The paper is structured as follows. First, a literature review is provided on ML-based process parameter prediction and adaptive scheduling. Then, in Section 3, the given problem in question is specified with the description of the analyzed production environment and the available/collected data. In Section 4, the proposed adaptive scheduling is described in detail and its efficiency is demonstrated by numerical experiments in Section 5. Finally, a conclusion and an outlook are provided in Section 6.

2 LITERATURE REVIEW

This paper focuses on adaptive scheduling through the ML-based prediction of remaining processing times. For this reason, it is investigated, whether there are already existing efforts that use the information from a ML-model on a process level for adaptive scheduling. In a first step, literature is reviewed regarding ML-based prediction of process parameters. The second step comprises efforts in the field of adaptive scheduling.

2.1 ML-based process parameter prediction

A large overview on existing application studies shows the general potential of using ML-models in production environment [Köksal 2011]. In this study, it is emphasized that ML-models are mostly applied for process parameter prediction. According to [Kim 2018], most ML-studies deal with the tool condition followed by process parameter predictions.

[Saadallah 2018] trained an ensemble of deep learning (DL)-algorithms to predict the stability of a milling process. Furthermore, random forests with 10,000 trees were applied for a condition monitoring system in order to predict the tool wear [Khorasani 2015, Saadallah 2018; Wu 2018]. [Krishnakumar 2015] followed another condition monitoring approach: Decision trees as well as artificial neural networks were applied to classify cutting tool conditions. The classification accuracy (ACC) of both algorithms were about ACC = 90 % [Krishnakumar 2015]. In order to predict tool wear, [Venkata 2014] implemented a feed forward neural network. [Yuan 2017] predicted future deformation based on Bayesian learning methods. A dual mode predictive controller was developed to reduce the machining vibrations. Thus, the quality of the product surface could be improved. In these cases, ML-models are implemented retrospectively, which means that there is no real-time ML-model implementation. [Yuan 2017]

In addition to the efforts regarding process parameter prediction such as tool wear, there are also studies regarding the forecasting of remaining processing or operating times. These examples are mostly focused regarding predictive maintenance. According to [Zhang 2018], the remaining useful lifetime (RUL) of a turbofan engine is predicted based on a long short-term memory (LSTM) neural network, which is applied to track the system degradation. The learning task of the ML-model is a regression that predicts the remaining cycles of the turbofan before it needs to be maintained. The data set was an open-source data set provided by NASA [Saxena 2008]. Comparable results were shown by [Mathew 2017], who used the same data set and implemented ten different algorithms to predict the RUL of the turbofan. He found out that a decision tree based ML-models such as random forest and gradient boosting lead to better results than support vector machine (SVM) or deep learning models. [Gugulothu 2017] predicts RUL using time series embeddings based on recurrent neural networks. The paper focuses on practical challenges in data-driven RUL

estimation including noisy sensor, missing data, and lack of prior knowledge about degradation trends.

In addition to forecasting the RUL, there are efforts to predict the remaining processing time. In the domain of business process management, [Bolt and Sepúlveda 2014] propose an approach to predicting the remaining processing time based on query catalogs that store the information of process events in form of partial trace details. These partial trace details are used to estimate the remaining time of new forms in process execution. Generally, the efforts show the potential of linking the information at process level with the throughput time. However, this link is not realized in production yet.

In conclusion, ML is already applied in production in order to predict process parameters. However, examples are limited to predictive maintenance and the prediction of RUL instead of using predictions of remaining processing times for an adaptive scheduling. In addition, the ML-model implementation takes place retrospectively instead of in real time.

2.2 Adaptive scheduling

A recent trend in production management is that decision makers rather tend to seek for robust solutions, instead of the optimal ones. In industrial practice, production planning and scheduling require much efforts if it meant to be efficient [Stricker 2014]. Furthermore, creating production schedules is not only a matter of time, but also requires actual production data from various sources (ERP, MES). Additionally, it is a strict requirement that the input data of scheduling must represent the real processes and system behavior, instead of applying idealistic parameters [Becker 2016]. Otherwise, the garbage-in-garbage-out rule will hold, namely that unrealistic and obsolete parameters-based schedules will fail to work well in the execution phase. Therefore, the target level of related KPIs (stock levels, lead times, workloads) will never be realized.

In addition to the quality of the input data, the quality of the scheduling process itself is of crucial importance. As mentioned, robustness has a higher priority recently than optimality. Robustness in production planning involves refined approaches that aim at handling predictable or unpredictable changes and disturbances. These approaches respond to the occurrence of random events with reactive approaches [Pfeiffer 2007, Monostori 2007], or protect the performance of plans by anticipating to a certain degree the occurrence of uncertain events with proactive approaches [Tolio 2011, Herroelen 2004].

As for the former, providing the robustness of a schedule in a reactive way means that the schedule needs to be quickly and efficiently recalculated, in case a certain deviation is realized during the execution. Rescheduling actions are usually triggered by the realized lateness, in case a predefined threshold is achieved. Then, the "live" schedule needs to be refined by leaving the orders under execution (and possibly within a so-called frozen period) unchanged, as it is technologically required, and recalculating the schedule considering the remaining set of work orders. A general requirement towards rescheduling is to make it rapidly and with the least possible hurt of the original schedule [Vieira 2003].

Considering the requirements and challenges above, some advanced approaches exist already that are aimed at providing robust solutions, relying on trustful production parameters. These approaches are mostly based on production data that is collected from manufacturing execution systems, as the latest historical data regarding the order completions usually reflects the actual state of the

system. Applying data analytics and machine learning approaches on such data provides the option to obtain input parameters that will result in valid and feasible plans. In [Gyulai 2017], a robust planning approach is presented combining manufacturing execution systems (MES) and enterprise resource planning (ERP) data as input of DDM to calculate with the latest state of the system and the true cycle times of the orders. Then, the parameters are fed into simulation models to predict the possible future behavior of the system under various scenarios, generating a large set of data. In order to utilize this data in scheduling, simulation metamodels are built by using machine learning approaches, and these models are integrated in the scheduling application.

Other machine learning approaches are also defined to support robust production schedules. [Csáji 2008] presented a reinforcement-learning solution, where the effects of machine breakdowns and late completions in scheduling are learnt over time, as well as the optimal scheduling policy is calculated providing efficient schedules in a highly uncertain environment.

Summarizing the revised literature in the topics of robust scheduling and rescheduling, it can be concluded that machine learning and data-driven solutions are proven to perform well in scheduling tasks. However, most of the approaches remain in a prototype phase without any transfer to real industrial environments. Therefore, production engineers and planner still face the challenges related to the robustness of the scheduling, in form of unexpected levels of key performance indicators (KPIs). Although, there are publications trying to link efficiently the execution and calculation stages of schedules, only a few efforts were made to combine them to schedule adaptively in production environment.

3 PROBLEM STATEMENT

Based on the literature review, the problem statement can be specified. In an ideal job-shop production environment, scheduling would lead to optimal production processes in terms of makespan. However, production environments are rarely ideal in reality. As process conditions often change, schedules should also be adjusted (rescheduling) to meet the target values of performance indicators.

In the scheduling problems, the changing process conditions and product qualities lead to changing machining times. This phenomenon is often realized in longer work order durations, and in some cases, it can even lead to the creation of new work orders that should be considered when rescheduling the production.

Nowadays, ML-models are used to predict process parameters or characteristics of product quality. In many cases, the application of ML-models in production remain on a prototype level. Especially the prediction of remaining processing times based on current process conditions offers the potential for adaptive scheduling and has not yet been investigated. It can be concluded that scheduling can be optimized using ML-models to increase the efficiency and flexibility of the overall production system.

4 ADAPTIVE SCHEDULING THROUGH ML-BASED PROCESS PARAMETER PREDICTION

The target architecture of the adaptive scheduling is shown in Fig. 1. In the very beginning, a schedule for jobs is calculated. The architecture takes the real-time adjustment of the schedule based on certain occurrences during machining into account.

Since the resource usage for the orders to be planned is the same, rescheduling basically affects all processing operations that have not yet been started. The problem of rescheduling can be determined as follows. When receiving ML-based predicted changes in processing times (order duration), the schedule for the set of unreleased orders must be recalculated according to the changed process parameters. Therefore, the target architecture is structured in the ML-based process parameter prediction, in which the remaining processing time per job is predicted and in the adaptive scheduling, which uses the information of the ML-model in order to adjust the schedule instantly.

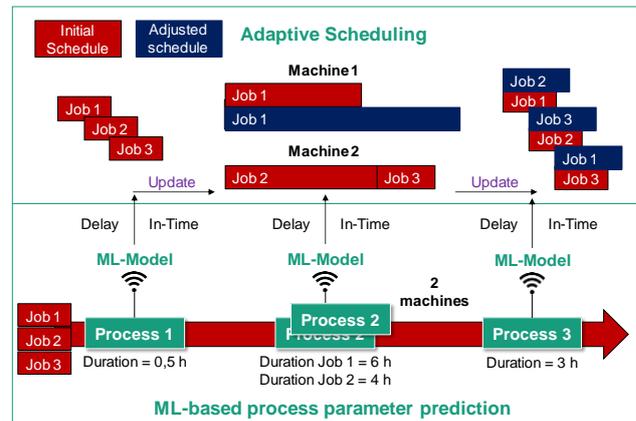


Fig. 1: Target architecture of adaptive scheduling based on ML-based process parameter prediction.

In order to enable adaptive scheduling in practice, two different scenarios are considered. These two scenarios are related to the match of the actual process time with the ideal one.

- Process is in-time
- Process is delayed through adjustments of process parameters

Process parameters can be adjusted during machining in case of critical process conditions. Therefore, it is assumed that the adjustments are made if critical thresholds of process parameters are exceeded.

4.1 ML-based process parameter prediction

In order to carry out DDM-projects successfully, methodological approaches are used in practice. The most often used methodology is the *cross-industry standard process for data mining (CRISP-DM)*, which is depicted in Fig. 2 [Chapman et al. 2000]. The first step comprises the understanding of the corresponding business. In this case, the manufacturing processes and interdependencies need to be understood.

Afterwards, data is initially acquired. Here, the focus is put on the acquisition of internal machine data, which includes axis data as well as data about the tool and motor currents. External sensors are used in order to acquire further information e. g. about accelerations and forces on the workpiece.

In the third step, the data is prepared for further modelling. The characteristics of the data sets are dependent on the protocol of the sensors as well as the acquisition concept (e. g. technically needed minimal frequency of time series). Since the data should be analyzed in real time and the information of ML-models should be fed towards the scheduling, a real-time system for communication needs to be deployed.

According to [Perwej 2018] and [Syafudin 2018], stream processing technologies such as Apache Flink or Kafka are

examples of platforms that enable real-time data processing. Stream processing differs from batch processing in a sense that data is processed directly at the time of generation [Ounancer 2017]. Apache Flink, e. g., processes data streams with very high volume at very low latency, because it is able to scale calculations to a large number of cores [Perwej 2018].

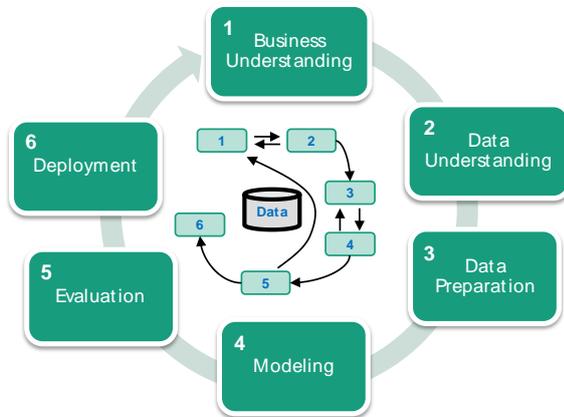


Fig. 2: Cross-Industry Standard Process for Data Mining (CRISP-DM) in the style of [Chapman et al. 2000].

Based on the real time acquisition, data needs to be processed. This includes steps such as the integration of data from different data bases into one data base, cleansing of missing values or duplicates and transformation of data formats. The preprocessing of data has a big influence on the performance of the ML-model in the end [Shopov 2013]. Before the fourth step of the CRISP-DM is performed, suitable ML-algorithms need to be selected based on the use-case requirements and data set characteristics. Furthermore, ML-algorithms have to be chosen related to the learning task. In order to consider the two different scenarios, a regression task is selected, in which the remaining processing time is predicted based on current process parameter values. Based on the remaining machining time, it can be identified whether the process is delayed or finished in-time through the comparison of the predicted and the assumed processing time. Fig. 3 illustrates the ML-model and output variable. Based on the remaining processing time, which is the output variable of the ML-model, the difference between the predicted and assumed processing time in schedule is calculated. Through the calculation, the delay time is identified that is in the best case $t_{delay} = 0$ min.

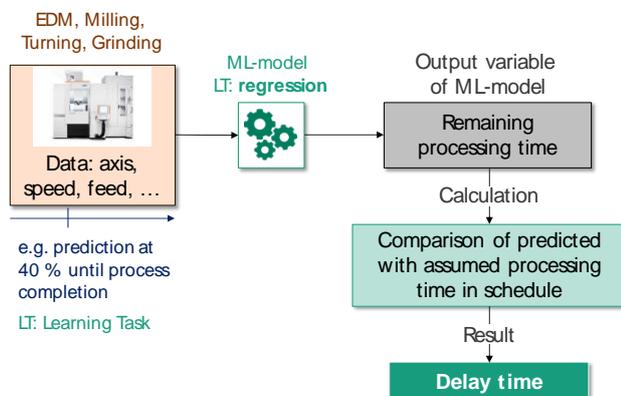


Fig. 3: Output variable of ML-model and scenario realization.

The learning task limits the number of suitable ML-algorithms. Possible ML-algorithms range from artificial

neural network (ANN) over decision tree-based algorithms such as random forest or gradient boosting to SVM. Open source technologies such as Python or R can be used for the final implementation of selected ML-algorithms. In order to enhance the performance of the ML-algorithm, hyperparameters need to be tuned. The resulting ML-model is trained based on a training data set. Therefore, the acquired data set needs to be split into the training and a test data set. The trained ML-model is tested based on a validation technique, which can be a k-fold cross validation or random subsampling and uses the test data set.

The results of the model are evaluated in the following step of CRISP-DM. For the learning task regression, suitable metrics range from mean square error (MSE) over root mean square error (RMSE) to mean absolute error (MAE). If the performance of the ML-models meet the requirements, the models can be deployed in the real production environment. Moreover, it needs to be ensured that the ML-model is working stable with the real-time communication system.

4.2 Rescheduling strategy

In order to reduce the negative effects of late rescheduling, e. g. high level of work-in-progress, idle machines and high overall lateness, the rescheduling needs to be performed at the earliest possible time when its necessity is recognized. In this case, a prediction from the side of the ML-models is received by the scheduler, a rescheduling event is triggered immediately in order to benefit most from the foresight on the remaining processing times. In general, the rescheduling action needs to consider several constraints and possible decision alternatives when generating the new schedule.

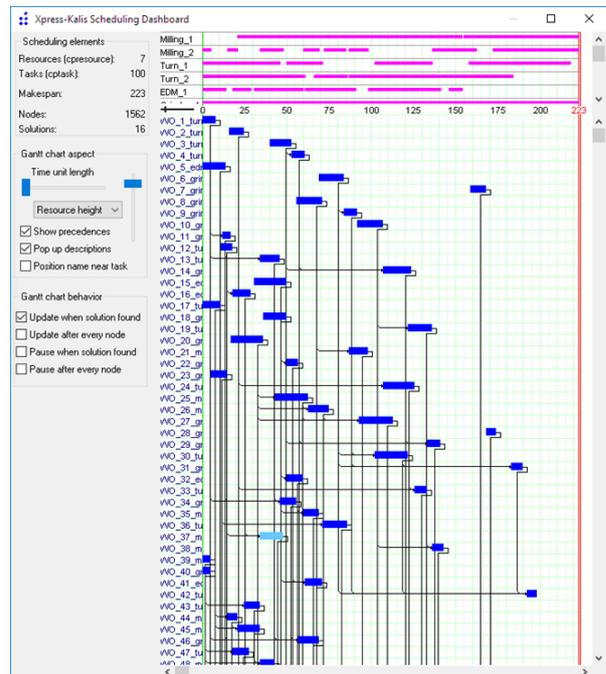


Fig. 4: Xpress-Kalis Scheduling dashboard.

As mentioned earlier, rescheduling aims at changing the original schedule in the least possible extent, in order to reduce the additionally generated changes, e. g. logistics and administrative efforts. For this reason and also for technological constraints, only those jobs can be considered within the rescheduling that are neither yet released, nor booked to be released at any machines. Therefore, a strict frozen period is to be considered, within a schedule cannot be changed. Within the experiments, it is assumed that all other work orders can be moved freely

in the schedule, considering time and resource assignments. An exemplary scheduling scenario of is depicted in Fig. 4 applying the Kalis constraint satisfaction solver provided by FICO Xpress.

Within a rescheduling, theoretically, three main options need to be considered based on the feedback information from the ML-results: The deviation of the remaining processing time from the ideal, the recognized quality issues that lead to rework and those that lead to scrap parts. In this paper, the first case is focused, if the existing set of work orders remains unchanged, however, the duration of the orders may change according to the prediction, so as influencing the subset of orders that will be released later in time.

5 NUMERICAL EXPERIMENTS

In this chapter, the scheduling model is formulated and the test scenarios described in detail. Afterwards, the results are presented and discussed.

5.1 Formulation of the scheduling model

As mentioned, this paper presents a proof-of-concept that DDM can increase the efficiency of the overall production system through adaptive scheduling by applying ML-models to predict remaining processing times. For this reason, a DES-model demonstrates and represents an overall machinery system in order to provide a sufficient set of numerical results. The DES-model provides the opportunity to link the scheduling algorithm with the prediction results, and analyze the efficiency of the early rescheduling.

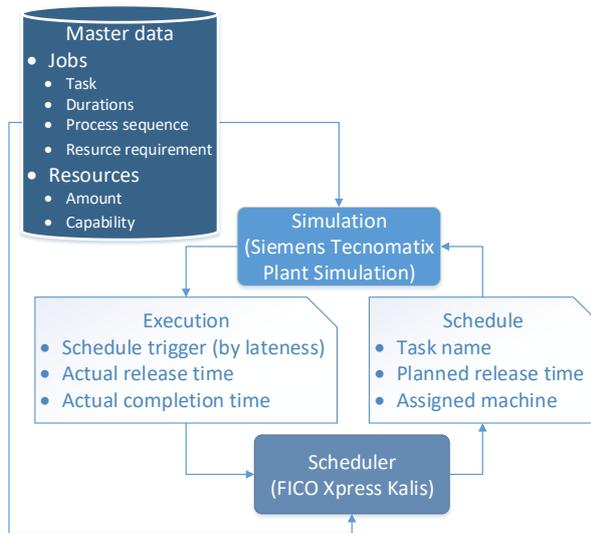


Fig. 5: Architecture of the application used in the experiments: synchronization and link of the scheduler and simulation models.

In this paper, a job-shop machinery is under study where four different machine types are available (milling, turning, grinding, electrical discharge machining, and alternative machines can be used to complete the given processes. From the scheduling viewpoint, a set of jobs is given which need to be completed by the available resources. Each job has a set of machining operations and the order of performing the operations is defined by the fix process plan. The machining times per operation are also part of the process plan and considered when scheduling the jobs. In general, the scheduling means the assignment of operations to machines and calculating the release times of the orders, considering the sequence of the operations

(precedencies), and the set of alternative resources as constraints (resources). In the presented case, the objective is to minimize the makespan of a schedule, considering a given set of jobs.

The scheduler application for the experimental purposes is implemented in FICO Xpress, which is a high-end mathematical solver with a constraint programming (CP) library called Kalis (Fig. 5). The scheduling model is formulated as a CP-model and solved by the default branching strategy of Kalis. The considered problem, in general, is a job-shop scheduling with alternative machines: having a set of machines M and a set of jobs N , and each job has corresponding work orders. The set of work orders is denoted by O , and each order has a given duration t_o , and a resource requirement m_o . The set of process (and also machine) types is denoted by S , each task has its own process type $s_o \in S$ and machines have process capabilities $s_m \in S$. The machines are considered as disjunctive resources, therefore, they are capable to process a single work order at any point of time. The objective function minimizes the makespan T , which means the overall completion time of all jobs. Considering the process plans of the jobs, the schedule includes some precedence constraints that prevent to hurt the predefined sequence of order completions. The precedencies between tasks are represented by a graph $G(V,E)$ with arcs $(i,j) \in E$ symbolizing that operation i precedes operation j . The binary decision variables a_{om} determine if order o is assigned (1) to machine m or not (0). The scheduling CP-model is formulated as it follows:

$$\text{minimize } T \quad (1)$$

$$t_o^{\text{end}} \leq t_o^{\text{start}} + t_o \quad \forall o \in O \quad (2)$$

$$t_o^{\text{start}} \leq t_p^{\text{start}} + t_o \quad \forall o, p \in O: (o, p) \in E \quad (3)$$

$$T = \max_{o \in O} t_o^{\text{end}} \quad (4)$$

$$a_{om} \in \{0,1\} \quad \forall o \in O, m \in M \quad (5)$$

$$\sum_{m \in M} a_{om} = 1 \quad \forall o \in O \quad (6)$$

$$\sum_{o: (t_o^{\text{start}} \leq t) \wedge (t_o^{\text{end}} \geq t)} a_{om} \leq |M| \quad \forall m \in M, t < T \quad (7)$$

$$a_{om} \cdot s_o = a_{om} \cdot s_m \quad \forall o \in O, m \in M \quad (8)$$

The objective function (1) expresses the minimization of the makespan of the schedule. Constraint (2) specifies the duration of the tasks, while (3) represents the precedence limitations. Equation (4) defines the makespan, and constraints (5-8) defines the machine capacities considering alternative resources.

5.2 Description of the test scenarios

As an initial part of an ongoing research, this paper presents only the very first results achieved in adaptive scheduling, focusing on the assessment of its potentials in a machining environment. In the test scenarios, input data was generated with random parameters, based on some real use-cases. The test environment is data-driven in a sense that the simulation model of the manufacturing environment and the scheduling model are parametric, therefore, various use cases can be analyzed. The use cases can differ in any of the parameters used in the planning model, however, for the sake of clarity, the same production environment in all test scenarios is used and varied only the set of jobs and the corresponding attributes.

The production environment considered in the experiments is characterized with the following parameters. The total number of machines is $|M| = 7$, including four different machine types that covers the whole set of machining

operations. In the experiments, milling, drilling, grinding and electric-discharging operations are assumed, which require different machine types to be performed. The number of jobs is set to be $|J| = 15$, each job has several operations in the range of 1 to 10 and the overall set of operations is $|O| = 60$. The durations of the operations ranged between $t_o = [10; 50]$. Around 30 – 40 % of the operation are set to have some lateness so as resulting longer processing times, with 10 – 40 % per job. The lateness is randomly generated by following a uniform distribution. Along the experiments, it is assumed that the ML-model can predict the total amount of remaining processing at 50 % of the (ideal) cutting process. Therefore, a rescheduling event can be triggered at the earliest time of 50 % operation completion. In the scheduling model, the algorithm was performed in most of the times for 10 seconds after a feasible schedule was found. In total, 10 test scenarios with different job and corresponding order attributes were defined.

5.3 Results of numerical experiments

The results of the test scenarios are summarized in *Tab. 1*. In the first column, the "ideal" case provides the calculated makespan of the original schedule (without rescheduling), assuming that all conditions are ideal tasks finished always on time. In the second column, the makespan is resulted by the execution of the schedule with simulation by adding late process completion events as described above. By nature, this case is always worse than the ideal one. The results in the last column (adaptive) includes the execution makespan values resulted by the proposed, adaptive rescheduling method based on the ML-based predictions. In general, the last two columns were examined during the experiments. The shorter makespan values are considered to be better, indicating that the corresponding scheduling method is less prone to the late task completions.

Tab. 1 : Makespan (T) results of the test scenarios.

Scenario [#]	T (ideal) [min]	T (late) [min]	T (adaptive) [min]	ΔT (diff.) [%]
1	329	458	433	-8%
2	398	517	499	-5%
3	538	604	560	-8%
4	355	425	393	-9%
5	358	478	457	-6%
6	421	480	436	-10%
7	308	444	479	10%
8	309	392	410	6%
9	439	535	472	-14%
10	283	407	290	-41%
AVG	378	474	442	-9%

Observing the very first test results of the ongoing research, the adaptive scheduling approach results in shorter execution makespan in the analyzed cases. In eight of the ten production scenarios, the adaptive scheduling provided an approximately 9% shorter execution makespan in average, indicating that it is worth the efforts to make further investigations in this direction, including many diverse production scenarios. Important to highlight is that a stopping criteria of 30 seconds was set in the scheduler, therefore, the scenarios #6 and #7 might differ in the makespan results due to the fact that the algorithm was terminated by a time limit, instead of an optimality.

6 CONCLUSIONS AND OUTLOOK

In this paper, a novel approach how to schedule the production order adaptively on the basis process parameter prediction was presented. In this context, ML-models are trained to forecast the remaining processing time in order to determine whether the process is delayed. The implementation of ML-models is conducted based on a methodological approach, in which data is acquired and preprocessed before ML-models are selected, implemented and optimized. The finally evaluated and deployed ML-model is able to predict remaining processing time with sufficient accuracy. This information serves as an input for an adaptive scheduler, which adjusts the schedule based on the current process information. The results of the numerical experiments show the potential of this approach. The cycle schedule execution makespan could be reduced by approximately 10 %, compared to the case when adaptive, ML-based scheduling was not applied. In terms of the scheduler model, the ML-model output was assumed as a randomly generated lateness.

The presented potential analysis of the application is the first step of an ongoing research. The authors will extend the set of experiments and detailed validation as well as verification of the results. Many experiments and enhancement of every module are envisioned in order to assess the complete potentials of the method, and also to identify the most promising use cases. Furthermore, within future experiments, various production environments are planned to be analyzed from the scheduling viewpoint, in order to find those that can be targeted by the adaptive scheduling approach with the highest expected benefits.

Further investigation will also focus on the process parameter prediction. In order to ensure a ML-based adaptive scheduling in practice, stable systems must be developed that reliably predict remaining processing time in real-time. Therefore, a real use-case is taken and a ML-model is trained with the output variable remaining processing time.

7 ACKNOWLEDGEMENTS

The research in this paper was partially supported by the European Commission through the H2020 project EPIC (<https://www.centre-epic.eu/>) under grant No. 739592. The Hungarian authors acknowledge the support by the ED_18-2-2018-0006 grant on a "Research on prime exploitation of the potential provided by the industrial digitalisation".

8 REFERENCES

Paper in a journal:

- [Bughin 2017] Bughin, J., Hazan, E., Ramaswamy, S., Chui, M., Allas, T., Dahlström, P., Henke, N., Trench, M. Artificial Intelligence – The Next Digital Frontier?, June 2017, McKinsey Global Institute
- [Csáji 2008] Csáji, B. Cs., and Monostori, L. Adaptive stochastic resource control: a machine learning approach, 2008, Journal of Artificial Intelligence Research, 32, pp. 453-486.
- [Gyulai 2017] Gyulai, D., Pfeiffer, A., Monostori, L. Robust production planning and control for multi-stage systems with flexible final assembly lines, International Journal of Production Research, 2017, 55:13, pp. 3657-3673
- [Herroelen 2004] Herroelen, W. and Leus, R. Robust and reactive project scheduling: a review and classification of procedures, 2004, International Journal of Production Research, 42:8, pp. 1599-1620.

- [Khorasani 2015] Khorasani, A. & Yazdi, M.R.S. Development of a dynamic surface roughness monitoring system based on artificial neural networks (ANN) in milling operation. *The International Journal of Advanced Manufacturing Technology*, October 2017, 93, 1-4, pp.141-151, ISSN 0268-3768
- [Kim 2018] Kim, D.-H. et al. Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 2018, 5(4), pp. 555–568. ISSN 2288-6206
- [Köksal 2011] Köksal, G., Batmaz, I. and Testik, M. C. (2011) 'A review of data mining applications for quality improvement in manufacturing industry', *Expert Systems with Applications*, 38(10), pp. 13448-13467
- [Monostori 2007] Monostori, L., Kádár, B., Pfeiffer, A. and Karnok, D. Solution approaches to real-time control of customized mass production, 2007, *CIRP Annals*, 56:1, pp. 431-434.
- [Ounacer 2017] Ounacer, S., Talhaoui, M.A., Ardchir S., Daif, A., Azouazi, M. Real-time Data Stream Processing Challenges and Perspective. *IJCSI International Journal of Computer Science Issues*, September 2017, Volume 14, Issue 5, ISSN 1694-0814
- [Perwej 2018] Perwej, Y., Husamuddin, Majzoob K, O., Bedinekerim. A Comprehend The Apache Flink In Big Data Environments. *IOSR Journal of Computer Engineering (IOSR-JCE)*, February 2018, Volume 20, Issue 1, pp. 48-58, ISSN 2278-8727
- [Pfeiffer 20007] Pfeiffer, A., Kádár, B., Monostori, L. Stability-oriented evaluation of rescheduling strategies, by using simulation, 2007, *Computers in Industry*, 58:7, pp. 630-643.
- [Schröder 2016] Schröder, C., Schleppehorst, S., Rosemarie, K. Bedeutung der Digitalisierung im Mittelstand, 2016, *IfM-Materialien*, No. 244, Institut für Mittelstandsforschung (IfM) Bonn, Bonn
- [Tolio 2011] Tolio, T., Urgo, M., Váncza, J. Robust production control against propagation of disruptions, 2011, *CIRP Annals*, 60:1, pp. 489-492.
- [Venkata 2014] Venkata K., Murthy, B. and Mohan N. Prediction of cutting tool wear, surface roughness and vibration of work piece in boring of AISI 316 steel with artificial neural network, 2014, *Measurement*, 51, pp. 63-70
- [Vieira 2003] Vieira G. E., Herrmann, J. W. and Lin, E. Rescheduling manufacturing systems: a framework of strategies, policies, and methods, 2003, *Journal of Scheduling*, 6:1, pp. 39-62.
- [Wu 2018] Wu, D., Jennings, C., Terpenney, J., Kumara, S. and Gao, R. X. Cloud-based parallel machine learning for tool wear prediction, 2018, *Journal of Manufacturing Science and Engineering*, 140(4), 041005
- [Yuan 2017] Yuan, Y., Zhang, Y.Y., Wu, Y., Zhu, T., Ding, H. Bayesian Learning-Based Model-Predictive Vibration Control for Thin-Walled Workpiece Machining Processes, *IEEE/ASME Transactions on Mechatronics*, 22(1), pp. 509–520, ISSN 1083-4435
- [Zhang 2018] Zhang, J., Wang, P., Yan, R., Gao, R. Long short-term memory for machine remaining life prediction, *Journal of Manufacturing Systems*, 2018.

Paper in proceedings:

[Becker 2016] Becker J. M. J., Kádár B., Colledani, M., Stricker, N., Urgo, M., Unglert, J., Gyulai, D. and Moser, E. The RobustPlaNet Project: Towards Shock-Robust Design Of Plants And Their Supply Chain Networks, 2016, *IFAC-PapersOnline*, 49, pp. 29-34.

[Bolt and Sepúlveda 2014] Bolt, A., Sepúlveda, M. Process remaining time prediction using query catalogs, *International Conference on Business Process Management*, 2014, *Business Process Management Workshops*, pp 54-65

[Krishnakumar 2015] Krishnakumar, P., Rameshkumar, K. and Ramachandran, K.I. Tool Wear Condition Prediction Using Vibration Signals in High Speed Machining (HSM) of Titanium (Ti-6Al4V) Alloy, 2015, *Procedia Computer Science*, 50, pp. 270–275

[Mathew 2017] Mathew, V., Toby, T., Singh, V., Rao, M., Goutham, K. Prediction of Remaining Useful Lifetime (RUL) of Turbofan Engine using Machine Learning. *IEEE International Conference on Circuits and Systems (ICCS)*, 2017

[Saadallah 2018] Saadallah, A., Finkeldey, F., Morik, K. and Wiederkehr, P. Stability prediction in milling processes using a simulation-based machine learning approach, 2018, *Procedia CIRP*, 72, pp. 1493-1498

[Shopov 2013] Shopov, V., Markova, V. Impact of Data Preprocessing on Machine Learning Performance, *Proceedings of the International Conference on Information Technologies (InfoTech-2013)*, September 2013, Bulgaria.

[Stricker 2014] Stricker N., Lanza G. The concept of robustness in production systems and its correlation to disturbances, 2014, *Procedia CIRP*, 19, pp. 87-92.

Technical reports or thesis:

[Chapman 2000] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. *Crisp-dm: Step-by-step data mining guide*

[Gugulothu 2017] Gugulothu, N., TV, Vishnu, Malhotra, P., Vig, L., Agarwal, P., Shroff, G. Predicting Remaining Useful Life using Time Series Embeddings based on Recurrent Neural Networks, 2017

[Saxena 2008] Saxena A, Goebel K, Simon D, Eklund N. Damage propagation modeling for aircraft engine run-to-failure simulation, *Int. conf. prog. heal. manag. PHM 2008*, <http://dx.doi.org/10.1109/PHM.2008.4711414>.

[Stein 2018] Stein, B. van. *Data driven modeling & optimization of industrial processes*, 2018

[Syafudin 2018] Syafudin, M., Ganjar, A., Norma, L.F., Jongtae, R. Performance Analysis of IoT-Based Sensor, Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing, September 2018