Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 93 (2020) 1025–1030

www.elsevier.com/locate/procedia

53rd CIRP Conference on Manufacturing Systems

Unsupervised Learning for Opportunistic Maintenance Optimization in Flexible Manufacturing Systems

Michael Wocker[a,b,*], Naomi Kimberly Betz[a], Christian Feuersänger[a], Alexander Lindworsky[a], Jochen Deuse[b]

[a]BMW Group, Petuelring 130, 80788 Munich, Germany
[b]Technische Universität Dortmund, Institut für Produktionssysteme, 44227 Dortmund, Germany

* Corresponding author. Tel.: +49-151-601-52564. *E-mail address:* michael.wocker@tu-dortmund.de

Abstract

Large scale manufacturing systems with a high degree in automation and the ability to produce several product variants in parallel meet current requirements of a highly flexible and at the same time productive manufacturing process. In practice, however, the non-transparency as well as the complexity of these systems overwhelm the maintenance department in the effective planning and implementation of maintenance tasks. As a result, major maintenance tasks are postponed to non-production times which causes increased maintenance cost as well as a decrease in system availability. This research explores a method that uses unsupervised learning algorithms to analyze type mixes and related process performances inside the system. The information is used to determine the optimal master production schedule prior to maintenance activities which leads to more frequent and extended time windows for maintenance activities during production time and thus to an increase in system availability.

Keywords: Flexible Manufacturing System; Opportunistic Maintenance; Unsupervised Learning; Type mix; Scheduling

1. Introduction

The automotive industry faces the challenge of providing a growing diversity of engine variants in times of volatile and competitive markets. This leads to an increasing demand for a flexible and at the same time productive manufacturing process [1,2]. A Flexible Manufacturing System (FMS) for mid- and high-volume engine-production offers the best trade-off with respect to high productivity and flexibility [3]. To realize a production rate of several hundred thousand units per year, the system consists of processing stations with multiple redundant machining centers that can process a variety of product variants. FMSs are built to "exploit the benefits of the Economy of Scope while achieving the efficiencies of the Economy of Scale" [4]. In reality, the system availability as well as the performance however fall short of expectations [5] which is presented in more detail in the following subsection. To restore system availability to a competitive level, an effective maintenance planning is crucial [5,6]. The proposed, data-driven approach contributes to the concept of Opportunistic Maintenance (OM), which is about cost-effective time windows for maintenance during production. The presented research for the optimization of a maintenance opportunity window (MOW) applies clustering algorithms to real production data to determine the varying throughput of processing stations given different type mixes as their input. The knowledge about process performance in relation to produced type mixes can be used to adopt the production schedule and to improve the MOWs for an effective maintenance. The approach is based on the CRISP-DM (CRoss-Industry Standard Process for Data Mining) that divides a data analytics project into six phases such as business and data understanding, data preparation, modeling, evaluation and the deployment. Shearer [7] provides additional background to the CRISP-DM.

10.1016/j.procir.2020.04.025

This paper is structured as follows. First, the main characteristics of the considered manufacturing system are described and the problem as well as the requirements for the method developed are defined. Chapter 2 summarizes existing approaches for OM in FMS and describes the fundamentals of the applied clustering algorithms. Subsequently, in chapter 3, the development of the method as well as the application results are presented. Chapter 4 summarizes the proposed solution and discusses further research issues.

1.1. Complexity in Flexible Manufacturing Systems

An FMS has multiple aspects of flexibility such as machine flexibility, material handling flexibility, process and product flexibility as well as volume flexibility [8,9]. The extend of these flexibility dimensions as well as the scaling of the system varies depending on production requirements. The research of this paper focuses on high-volume FMSs with numerous processing steps. The processing stations P_n are interconnected by an automated material handling system and decoupled through the implementation of a buffer with variable capacity. A processing station consists of several redundant machining centers M_m which all perform the same processing step in parallel to increase the production volume. Each machining center is flexible and has an automized tool change which enables it to process various part types. Fig. 1 shows the conceptual layout of the FMS.

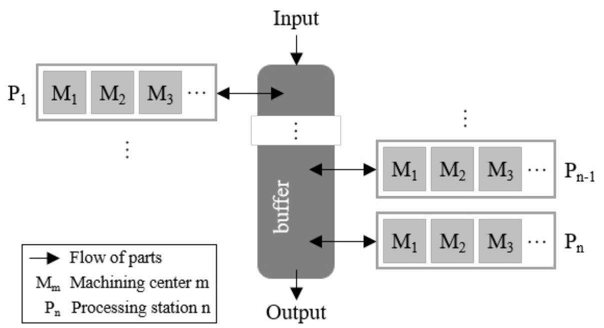

Fig. 1. Concept of the examined Flexible Manufacturing System.

The simultaneous production of part types with different cycle times combined with the flexible material handling system lead to a non-transparent material flow. Variable buffer capacities and an increased amount of random failures due to the high degree of automation cause fast changing bottlenecks and increase the complexity of the system [10].

1.2. Problem definition and requirements

The system's dynamics, like rapidly changing bottlenecks, as well as the lack of knowledge due to non-transparent processes make it difficult for the maintenance department to plan maintenance tasks in a way that minimizes their impact on system performance. As a result, maintenance tasks are either conducted causing unexpected production losses or they are postponed to non-production periods. Postponement leads to a less efficient use of the maintenance personnel capacities during production time and thus to additional labor costs due to extra shifts of maintenance during the non-production period. Eventually, the postponement of maintenance measures causes an increased risk of failure and thus a lower system availability. The strategy of OM, which is essentially the dynamic search for non-critical time windows for maintenance activities during production, is a promising concept to address this issue [11] and leads to the first requirement. Secondly, the method required has to be applicable in an FMS as presented above. The high degree of automation needed to manage the flexible material flow makes the FMS a cyber physical system and sets the prerequisite for a comprehensive data base [10]. Analyzing this database with big data technologies offers a huge potential to support the cognitively overwhelmed personnel in FMS [12]. Consequently, the use of real data to develop a method for an effective maintenance planning sets the third requirement for this research.

2. State of the art and research

The term "Opportunistic Maintenance" first appeared in publications by McMill and Radner as well as Jorgenson in 1963 [13,14]. According to the survey of Ab-Samat, the research interest for OM increased continuously in recent years but is still at the beginning and thereby full of potential [15]. There are two dimensions of OM which need to be looked at separately. First, OM can be defined as a policy for the replacement of one part or several parts at the same time (opportunity), given the condition of the other parts [14]. The other dimension, which defines the scope for this paper, considers the system as a whole and aims to find MOWs in which a maintenance activity in general can be executed without adverse effects on production goals [16].

2.1. Opportunistic Maintenance Methods for Flexible Manufacturing Systems

Gu et al. [17] mathematically described a basic configuration of a "two machines, one buffer" system that can be combined to larger and more complex systems. The MOWs are calculated using a simulation model that is fed with current buffer states. In addition to buffer states, Chang et al. [18,19] also investigated the failures of surrounding machines to determine MOWs in a transfer line using a simulation approach. Both papers consider systems that produce only one product and therefore fail to meet the requirement of an FMS. Furthermore, the use of simulation in FMS to gain highly dynamic information, such as MOWs is a complex and cost intensive method that might not lead to the desired accuracy [20,21]. Wocker et al. use real data to train a prediction algorithm that estimates MOWs in an FMS and outlines the proactive optimization of these MOWs using an appropriate production schedule [11]. In the context of production scheduling, further research exists that does not explicitly address OM and MOWs but whose methods and goals relate to the idea of this paper. Feng et al. [22] investigated different scheduling approaches to analyze the impact on system throughput in general. According to Sawik [23], cyclic scheduling is the best approach for maximizing the throughput of a Multi-Job Production system. Alavian et al.

mathematically analyzed the system throughput as well as system bottlenecks as a function of the product-mix. The findings lead them to "a so-called Product-mix Performance Portrait, which represents the system behavior for all feasible product-mixes and which can be used for operations management and improvement" [24]. Other than required by this approach, Alavian et al. consider a system with fixed material flow (first in first out). Furthermore, the approach does not use real data but theoretical assumptions that, for the application in a rather simple system with 11 machines and two variants, lead to an error in throughput calculation with an average of 5.1%.

On the one hand, existing approaches for OM in FMS focus on the given condition of the system but do not consider the potential of the possibility to adapt the production schedule for MOW optimization. On the other hand, current research on production scheduling and sequencing lacks the use of real data and does not treat the MOW of individual machines as a criterion for optimization. Using clustering algorithms to automatically determine the most relevant produced type mixes together with the corresponding process throughput provides promising information for a scheduling approach that optimizes maintenance in FMS with multiple products.

2.2. Clustering Algorithms

In general, clustering algorithms are unsupervised machine learning techniques that support the extraction of latent data structures from a given unlabeled data set. In order to find a convenient, structured organization of the data, clustering methods classify each data sample into a previously identified group, a so-called cluster. Following the definition of Jain and Dubes [25], "a cluster is comprised of a number of similar objects collected or grouped together". The underlying characteristics whether samples belong to the same cluster or not differ depending on the applied clustering technique.

The variety of clustering techniques available can be categorized into hierarchal and partitional approaches [26]. Given this paper's objective, the large amount of data and its high degree of dimensionality, hierarchical approaches are not considered due to their high time complexity. From partitional approaches, the most commonly applied clustering models have been implemented, i.e. *k*-means (centroid-based), Expectation-Maximization (EM) algorithm (distribution-based), mean shift clustering and DBSCAN (both density-based). [27,28]

The following paragraphs provide a concise overview of these four clustering algorithms and their application.

2.2.1. k-means

k-means is a widely used non-probabilistic clustering technique, which assigns each data point $\boldsymbol{x}_i \in \mathbb{R}^D$ to one of k circular clusters based on their Euclidean distance, where the number of clusters k must be determined a-priori [28].

The location of each cluster k is defined by its centroid $\boldsymbol{m}_k \in \mathbb{R}^D$. The method then maps each data point $\boldsymbol{x}_i$ to one of the k clusters such that the distance between all data points and their respective cluster centroids are minimized. Therefore, the iterative *k*-means algorithm from Lloyd [29] is applied, which performs an alternating optimization of data point classification and the cluster centroid locations $\boldsymbol{m}_k$.

First, the cluster centroids $\boldsymbol{m}_k$ are randomly initialized. Second, each data point $\boldsymbol{x}_i$ is assigned to its nearest cluster, i.e. the cluster whose centroid $\boldsymbol{m}_k$ has the shortest distance to the respective data point. Then, based upon the classification of all data points, the cluster centroid location $\boldsymbol{m}_k$ is updated by taking the mean of all data points assigned to its cluster, which in turn affects the classification of data points. These iteration steps are repeated until $\boldsymbol{m}_k$ converges, i.e. until no changes occur in either the cluster indicator or the cluster centroid location after one iteration step. [29,30]

2.2.2. EM Algorithm for Gaussian Mixture Models

Provided that the data approximates a Gaussian distribution, the clusters are allowed to take different forms than pure circles. In a Gaussian Mixture Model (GMM), the shape of each cluster is defined via the mean and standard deviation of the underlying data, which allows each cluster to take on any form of ellipse [28]. This approach is probabilistic as the component density parameters maximize the likelihood of a sample. The two parameters of each Gaussian cluster, i.e. the mean μ_k and covariance Σ_k, can be specified by applying the EM algorithm as introduced by Dempster et al. [31].

This concept suggests a simple scheme of alternative iterations until the specified error converges. First, the model parameters are initialized with random values. Then, the algorithm iteratively updates the probability of each data point belonging to a particular cluster, i.e. the so-called responsibilities, and the model parameters in what is called the E- and M-Step. The E-step, or expectation step, evaluates the responsibilities based on the current parameter values. During the maximization step, or M-step, these responsibilities are used to reevaluate the model parameters of each cluster, i.e. the means, covariances and mixing coefficients. [28,31]

2.2.3. Mean shift

Introduced by Fukunaga and Hostetler [32] in 1975, the basic principle of the non-parametric clustering algorithm is the interpretation of the feature space of the parameter as an empirical Probability Density Function (PDF) using a generalized kernel approach, where the bandwidth h defines a kernel's window radius. The distribution modes, i.e. cluster centroids, are space regions of high density in the Euclidean space and hence cause local maxima in the PDF. The location of each mode identified in the underlying distribution can be determined by performing a gradient ascent procedure for each data point $\boldsymbol{x}_i$ until convergence, i.e. the gradient of the density function is equal to zero. The location at which the gradient equals zero represents the stationary positions of the cluster centroids. All data points that converge to the same stationary point can then be grouped into the same cluster. [33,34]

2.2.4. DBSCAN

The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) uses, as the name implies, a density distribution as basis for the cluster estimation. The method starts the clustering process by visiting an arbitrary point $\boldsymbol{x}_i$ and retrieving all neighboring points within a predefined distance ε. If the number of retrieved points is sufficiently large, i.e. exceeds the specified threshold $N_{\min}$, a new cluster can be formed using the current data point as first cluster point. If not, the data point is considered to be noise, and the next arbitrary data point is considered. All points within the ε distance proximity of the first cluster point are considered part of the same cluster. The process of assigning all points in the vicinity of ε to the same cluster is repeated for all new points that have been allocated to the cluster group. Upon completion of the current cluster, a new unexplored data point is queried, resulting in either the discovery of another cluster or noise. This process is repeated until all points have been either assigned to a cluster or marked as noise. [35]

3. Method development and Evaluation

The clustering algorithms are applied to data that originates in a real FMS for the crankcase production of a car manufacturer. The explanation of the data preparation and preprocessing is followed by the presentation of the clustering results and its use for maintenance optimization in FMS.

3.1. Data Preparation and Preprocessing

The raw data record D consists of an event-based table with one row per production event (part leaves machine) and with the corresponding timestamp as its label. Moreover, attributes like serial number, machining type, processing time or processing events during the recorded production step are stored in D. The objective of the clustering analysis is to identify relevant type mixes of parts produced within a certain period of time. Accordingly, the underlying data set for the clustering model must also reflect possible type mixes. In order to agglomerate individually produced parts to type mixes, a sliding window method is introduced, see Fig. 2.

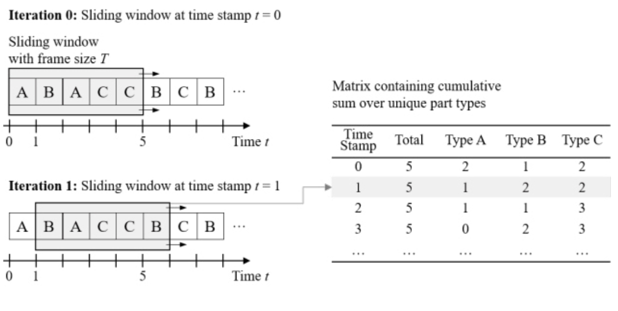

Fig. 2. Data preprocessing: Scheme of windowing function.

The sliding window method takes a data subset $W \subset S$ where S is a subset of the original data record D, filtered by a specific processing station. Subset W contains all data rows from S within a predetermined time period T which has to be chosen according the cycle time of the system. For W, the type mix can be determined by calculating the share of each part type in the total production of the considered processing station. In order to explore the development of the type mixes of the regarding station, the time window of size T iteratively slides through the record S, which is why this is called the sliding window method. Since all parts must pass through the initial station P_1, the dataset $S_1 \subset D$ filtered by P_1 is most suitable for the identification of type mix clusters. Fig. 3 shows the windowed type mix distribution based on dataset S_1 containing one week of production data. The frame size of the window was set to one hour.

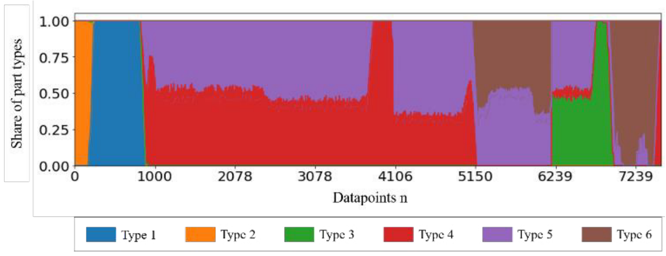

Fig. 3. Produced type mix for one week at initial process P_1.

During the considered week, six types of crankcases have been produced. A maximum of three different crankcases have been manufactured at the same time, see the range starting at datapoint 6239 in Fig. 3.

3.2. Results and Evaluation of the Clustering Algorithms

Applying clustering algorithms to the corresponding preprocessed dataset S_1 enables the automatic identification of common type mixes in large production data sets. Due to the varying properties of the proposed clustering methods (see subsection 2.2) the four clustering algorithms (a) k-means, (b) EM for Gaussian mixture models, (c) mean shift and (d) DBSCAN are implemented and their respective performances are compared. The clustering results for the data represented in Fig. 3 are shown in Fig. 4.

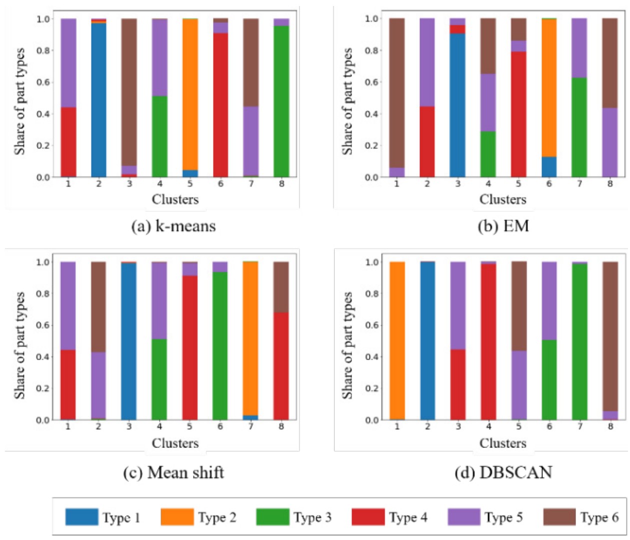

Fig. 4. Clusters of the investigated clustering algorithms.

The k-means and the EM algorithm are not able to automatically specify the optimal number of clusters k. In order to enable an automatic determination of k for both methods, the silhouette coefficient is utilized. The silhouette coefficient is a measure of the quality of a clustering which is independent of the number of clusters k [36]. The implementation of the two remaining clustering methods, mean shift and DBSCAN, requires the specification of one and two hyperparameters, respectively: the bandwidth h defining the kernel window radius in mean shift, as well as the two parameters ε and N_{min} specifying the neighborhood in DBSCAN.

By a visual comparison, it can be determined that the DBSCAN algorithm identifies the cleanest clusters, as displayed by the cluster centroids in Fig. 4.d. Since the algorithm is able to detect noise automatically, the transitions between type mixes are disregarded, which results in very cleanly separated clusters. The three outlier-sensitive algorithms are not able to distinguish between clear type mixes and transitions of type mixes, see e.g. transition from type 2 (orange) to type 1 (blue) in Fig. 3 and the respective shift of the cluster centroids in Fig. 4.a-c. Mean shift shows the least distortion followed by k-means and the EM algorithm. The clear distinction of clusters using DBSCAN can be observed in the 3D data representation obtained by applying a Principle Component Analysis (PCA). The PCA reduces the dimensionality and calculates eigenvectors (principal components) that lie along the axis of maximum variation in the data [37]. The DBSCAN algorithm confirms its ability to separate classes sharply by defining noise data points, as can be derived from Fig. 5.d.

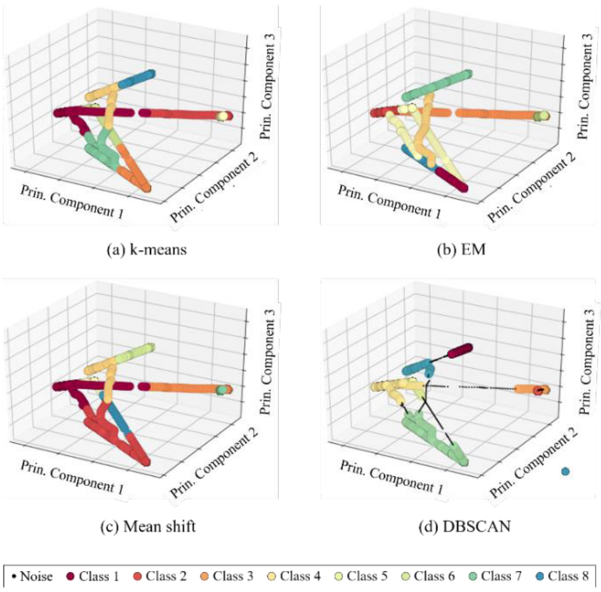

Fig. 5. PCA plot of the investigated clustering algorithms.

Nevertheless, the PCA again suggests that the EM algorithm, despite distorting the cluster centroids due to noisy data, effectively separates the data points of one class from those of other classes. In the respective Class 2 of the k-means (Fig. 5.a) and Class 3 of the mean shift (Fig. 5.c) algorithms, for example, data points that are positioned closer to the second or third Class, respectively, are assigned Class 1. Meanwhile, the PCA of the EM method in Fig. 5b shows that these data points are still included in Class 3. The observed results are evidenced in both smaller and larger datasets ranging from one day to one month, given an appropriate hyperparameter tuning for DBSCAN and mean shift. The high sensitivity of DBSCAN towards its parameters N_{min} and ε and as well as of mean shift towards its bandwidth h makes them not feasible for automated clustering in an industrial environment. Even minor changes in the setup or in the dimension of input data require re-tuning of the parameters since unnoticed changes in the surrounding conditions can induce large deviations in the results of the algorithm, rendering them unreliable in practice. On the other hand, further research reveals that by cleaning the data by performing a gradient ascent procedure and thereby deleting transitions between type mixes, the EM algorithm delivers as good of results as DBSCAN. Since EM algorithm is independent of hyperparameter tuning, it was used for further method development. By projecting the clusters calculated by the EM algorithm onto other processes datasets S_i and evaluating the corresponding throughput using the ratio parts per hour (pph), type mix specific throughput of a processing station can be evaluated. Since the overall availability of a processing station varies in time, the throughput is not comparable without scaling the data to a certain availability level which was chosen to be the average of the real system. Tab. 1 shows the process performance p of a random process per cluster with p being the quotient of the real processing station's throughput and the target throughput of the system. Numbers greater one indicate that the throughput of the processing station for an assumed availability of 95% is above the desired target throughput of the system. The maintenance Key Performance Indicator (KPI) T_{M} is calculated as

$$T_M = n \cdot \left(1 - \frac{1}{p}\right), \tag{1}$$

with n representing the number of machining centers of a processing station. Assuming that every machining center is set up for the same part types, T_{M} is a measure to determine the amount of machining centers that can be shut down for maintenance by still providing enough capacity to produce the target output of the system.

Tab. 1: Performance of a processing station and Maintenance-KPI per cluster.

Cluster	Process Performance p	Maintenance-KPI T_{M}
1	1.196	0.982
2	1.223	1.096
3	1.230	1.121
4	1.188	0.948
5	1.192	0.965
6	1.266	1.262
7	1.366	1.608
8	1.287	1.338

This information can be used for the proactive optimization of the MOW of a certain processing station. By scheduling the type mix of the cluster with largest value for T_{M}, the MOW of the considered processing station is extended. This results in less performance losses caused by maintenance. Furthermore, it gives the maintenance management the opportunity to more frequently schedule maintenance tasks during production time instead of postponing them into non-production periods. This allows the existing maintenance personnel capacities to be used more efficiently and the system availability increases as maintenance tasks are more often carried out on time.

4. Conclusion and future research

In this paper, unsupervised learning in the form of clustering algorithms has been investigated for its suitability of detecting major type mixes in real data of an FMS. With regard to an automated applicability in industry, the EM algorithm has performed the best among the tested algorithms when additional data preparation was done. If the application of the algorithm is supervised by an expert who constantly updates the hyperparameters, DBSCAN would be the algorithm of choice since it handles noise the best. The information about the found type mix clusters and the corresponding throughput have been used to calculate a meaningful KPI which allows the production schedule to adapt for maintenance optimization by extending maintenance opportunity windows.

The investigation of other clustering algorithms and further research into other FMSs needs to be conducted to prove stability and transferability of this method. Furthermore, in case of different setup states of machines belonging to the same processing station, the proposed KPI needs to be refined to express the impact of shutting down each machine individually. Since this paper established an initial approach and focused on the evaluation of the clustering algorithms and the development of a meaningful KPI, the optimization method needs to be applied and evaluated as part of operations and maintenance management during real production in an FMS.

References

[1] Ren W, Wen J, Guan Y, Hu Y. Research on assembly module partition for flexible production in mass customization. Procedia CIRP 2018;72:744–9.

[2] Ganschar O, Gerlach S, Hämmerle M, Krause T, Schlund S. Produktionsarbeit der Zukunft - Industrie 4.0: Studie. Stuttgart: Fraunhofer-Verl; 2013.

[3] Weck M. Werkzeugmaschinen 1. 6th ed. Berlin, Heidelberg: Springer Berlin Heidelberg; 2005.

[4] ElMaraghy HA. Flexible and reconfigurable manufacturing systems paradigms. Int J Flex Manuf Syst 2006;17(4):261–76.

[5] Koren Y. The global manufacturing revolution: Product-process-business integration and reconfigurable systems. Hoboken, NJ: Wiley a John Wiley & Sons Inc; 2010.

[6] Biedermann H. Anlagenmanagement: Managementinstrumente zur Wertsteigerung. 2nd ed. Köln: TÜV Media; 2008.

[7] Shearer C. The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing 2000;5(4):13–22.

[8] Sethi AK, Sethi SP. Flexibility in manufacturing: A survey. Int J Flex Manuf Syst 1990;2(4):289–328.

[9] Browne J, Dubois D, Rathmill K, Sethi S, Stecke K. Classification of Flexible Manufacturing Systems. The FMS Magazine 1984;2.

[10] Deuse J, Klenner F, Lenze D, Friedrich T. Flexible Fertigungssysteme der Generation I4.0 – Herausforderungen für Betreiber und Instandhalter. In: Biedermann H, editor. Lean smart maintenance: Konzepte, Instrumente und Anwendungen für eine effiziente und intelligente Instandhaltung: 30. Instandhaltungsforum, 1st ed. Köln: TÜV Media; 2016, p. 31–48.

[11] Wocker M, Wolf P, Lindworsky A, Deuse J. Opportunistische Instandhaltungsplanung in Flexiblen Fertigungssystemen. ZWF 2019;114(5):250–4.

[12] Schoeneberg K-P. Komplexitätsmanagement in Unternehmen: Herausforderungen im Umgang mit Dynamik, Unsicherheit und Komplexität meistern. Wiesbaden: Springer Gabler; 2014.

[13] McCall JJ. Operating Characteristics of Opportunistic Replacement and Inspection Policies. Management Science 1963;10(1):85–97.

[14] Radner R, Jorgenson DW. Opportunistic Replacement of a Single Part in the Presence of Several Monitored Parts. Management Science 1963;10(1):70–84.

[15] Ab-Samat H, Kamaruddin S. Opportunistic maintenance (OM) as a new advancement in maintenance approaches. J of Qual in Maintenance Eng 2014;20(2):98–121.

[16] Dekker R, van Rijn C. PROMPT, A Decision Support System for Opportunity-Based Preventive Maintenance. In: Özekici S, editor. Reliability and Maintenance of Complex Systems. Berlin, Heidelberg: Springer Berlin Heidelberg; 1996, p. 530–549.

[17] Gu X, Lee S, Liang X, Garcellano M, Diederichs M, Ni J. Hidden maintenance opportunities in discrete and complex production lines. Expert Systems with Applications 2013;40(11):4353–61.

[18] Chang Q, Ni J, Bandyopadhyay P, Biller S, Xiao G. Maintenance Opportunity Planning System. J. Manuf. Sci. Eng. 2007;129(3):661–8.

[19] Chang Q, Biller S, Xiao G. Transient Analysis of Downtimes and Bottleneck Dynamics in Serial Manufacturing Systems. Journal of Manufacturing Science and Engineering 2010;132(5):051015-1-9.

[20] Choi SH, Lee JSL. Computational algorithms for modeling unreliable manufacturing systems based on Markovian property. European Journal of Operational Research 2001;133(3):667–84.

[21] Ni J, Jin X. Decision support systems for effective maintenance operations. CIRP Annals 2012;61(1):411–4.

[22] Feng W, Zheng L, Li J. Scheduling policies in multi-product manufacturing systems with sequence-dependent setup times and finite buffers. International Journal of Production Research 2012;50(24):7479–92.

[23] Sawik T. Batch versus cyclic scheduling of flexible flow shops by mixed-integer programming. International Journal of Production Research 2012;50(18):5017–34.

[24] Alavian P, Denno P, Meerkov SM. Multi-job production systems: definition, problems, and product-mix performance portrait of serial lines. International Journal of Production Research 2017;55:1–26.

[25] Jain AK, Dubes RC. Algorithms for clustering data. Englewood Cliffs, NJ: Prentice-Hall; 1988.

[26] Jain AK, Murty MN, Flynn PJ. Data clustering. ACM Comput. Surv. 1999;31(3):264–323.

[27] Witten IH, Frank E, Hall MA. Data mining: Practical machine learning tools and techniques. 3rd ed. Burlington MA: Morgan Kaufmann; 2011.

[28] Alpaydin E. Introduction to machine learning. 2nd ed. Cambridge, Mass: MIT Press; 2010.

[29] Lloyd S. Least squares quantization in PCM. IEEE Trans. Inform. Theory 1982;28(2):129–37.

[30] Bishop CM. Pattern recognition and machine learning. New York: Springer; 2006.

[31] Dempster AP, Laird NM, Rubin DB. Maximum Likelihood from Incomplete Data Via the EM Algorithm. Journal of the Royal Statistical Society: Series B (Methodological) 1977;39(1):1–22.

[32] Fukunaga K, Hostetler L. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Inform. Theory 1975;21(1):32–40.

[33] Cheng Y. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Machine Intell. 1995;17(8):790–9.

[34] Comaniciu D, Meer P. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Machine Intell. 2002;24(5):603–19.

[35] Ester M, Kriegel H-P, Sander J, Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining 1996;96(34):226-231.

[36] Rousseeuw PJ. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 1987;20:53–65.

[37] Hastie T, Tibshirani R, Friedman JH. The elements of statistical learning: Data mining, inference, and prediction. 12th ed. New York, NY: Springer; 2017.