Proceedings IRF2018: 6th International Conference Integrity-Reliability-Failure
Lisbon/Portugal 22-26 July 2018. Editors J.F. Silva Gomes and S.A. Meguid
Publl. INEGI/FEUP (2018); ISBN: 978-989-20-8313-1

PAPER REF: 7283

CLUSTER ANALYSIS FOR ENHANCING PROCESS QUALITY IN JOB SHOP PRODUCTION

Antonia Fels[(*)], Max Ellerich, Robert Schmitt
Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Germany
[(*)]*Email:* a.fels@wzl.rwth-aachen.de

ABSTRACT

This work presents a clustering algorithm for finding an optimal set of taktlines in the context of synchronized job shop production. Based on jobs' frequently reoccurring process sequences, a clustering analysis is applied in order to find groups of process sequences, which can be aggregated with regard to an optimized material flow and shop floor logistics. To this end, a distance measure for process sequences is developed as well as a clustering algorithm based on partitioning clustering. Different scenarios will be discussed; industrial data from tool manufacturers serve as validation basis.

***Keywords*:** cluster analysis, job shop production, process quality, string alignment, machine learning

1. INTRODUCTION

The transfer of lean concepts to job shop production still bears huge potential. Besides digitalization and automation, the layout and organization of these production facilities is most relevant to its overall performance regarding process quality and productivity (Zwanzig, 2010). Traditional job shop production is characterized by a heterogeneous product program, where jobs usually follow individual paths through the shop floor, i.e. individual process sequences, while requiring non-deterministic processing times (Gruß, 2010). The individual process sequences render the material flow non-transparent and more generally the logistic effort very high, since each job passes the shop floor timely and spatially unbound from other jobs (Humphrey, 2016). In order to enhance the 'flow character' of a job shop production, a concept adapted from serial production has been brought forward in recent years: the synchronization principle (Gruß 2010, Ziskoven 2013). Latest concepts combine job shop production with the synchronized flow in terms of a shop floor wide 'takt', also well-known from serial production. Accordingly, jobs, which resemble in their process sequence, are bundled and transported jointly across the shop floor along so called 'taktlines'. The taktlines are to be understood as a series of taktstations, which comprise each of one or more taktsubstations, i.e. machines (Ziskoven, 2013; Humphrey, 2016).

Results from research and industrial practice support the effectiveness of synchronized job shop production to date (Humphrey, 2016). Whereas research in this field has examined the allocation of jobs to a given set of taktlines on the one hand (Schmitt, 2016), and addressed as well in detail the scheduling problem of synchronized jobs shop production (Ziskoven, 2013; Humphrey, 2016), the optimal layout of the taktlines system has solely been discussed on a conceptual level (Ziskoven, 2013). The layout of taktlines is in essence a combinatorial optimization problem, facing multiple contradicting objectives. The aim of this research is therefore the development of an algorithm for finding this optimal set of taktlines, considering restrictions and objective dimensions. To this end, a clustering approach is proposed, which considers the jobs' process sequences as strings; it can thus be viewed as a sequence-clustering problem.

The paper is organized as follows: The fundamentals of synchronized job shop production will be outlined in chapter two, resulting in the current research question. In chapter three, the proposed clustering analysis is outlined under consideration of the state of the art, leading to a presentation of results in chapter four. The paper closes with a discussion and outlook for further research.

2. SYNCHRONIZED JOB SHOP PRODUCTION

Traditional job shop production, which yields a highly heterogeneous product program, is characterized by its unstructured material flow and high logistic effort (Klotzbach, 2007). Large idle times and low machine utilization often result from a reactive manufacturing schedule (Günther, 2012). To overcome these insufficiencies, the transfer of so called *lean concepts* has been brought forward in recent years (Gruß, 2010). Besides product and process standardization (Klotzbach, 2007), especially the concept of synchronization in job shop production has been investigated. In this context, the work of Gruß, Zwanzig, Ziskoven and Humphrey are highly relevant. Whereas Gruß (2010) and Zwanzig (2010) examine the subject conceptually and thus lay the groundwork for synchronization in job shop production, Ziskoven (2013) and Humphrey (2016) specify the approach. They support the idea of bundling different jobs and transporting them simultaneously through manufacturing along so-called *taktlines* (Humphrey, 2016). A comparison of the material flow in traditional job shop production vs. synchronized job shop production is drawn in Figure 1.

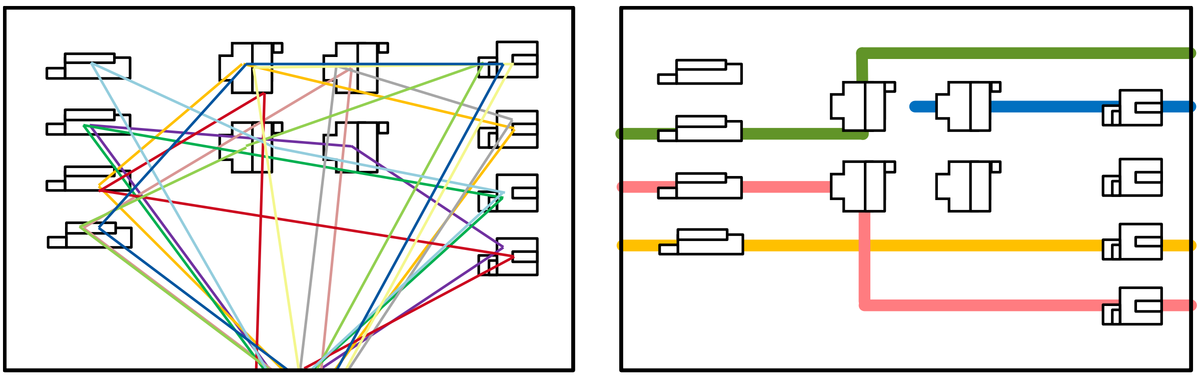

Fig. 1 - Comparison of traditional and synchronized job shop production (Humphrey 2016, p. 25)

The material flow according to traditional job shop production is condolidated in the right illustration. By defining a *limitied number of taktlines*, the transparency on the shop floor can be enhanced, logistic effort and idle time can be reduced (Ziskoven, 2013).

The implementation and operation of the synchronized job shop production usually follows three main steps, namely the layout of the taktline system, the allocation of incoming jobs to these taktlines and the actual (job shop) scheduling (Ziskoven, 2013). Whereas the layout of the taktlines system can be viewed as long-term, the allocation and scheduling are pursued on a daily or at least weekly basis (Humphrey, 2016). The objective criteria for the layout, implementation and operation of a synchronized production system correspond to well-known objective in production, i.e. minimization of through put time, low work in progress, high machine utilization and low operating costs (Humphrey, 2016; Ziskoven, 2013).

As mentioned in the introduction, the allocation and scheduling have hitherto been addressed, including the development of necessary algorithms. The layout of the taktline system on the other hand has not yet been investigated with regard to a mathematical optimization. Yet, in his work, Ziskoven states the general requirements for the layout of the taktlines system conceptually (Ziskoven, 2013).

An optimal layout of a production facility is, however, crucial for the overall target achievement of the manufacturing efficiency (Eversheim, 1999). Hence, a suitable set of taktlines allows for manufacturing a broad range of products according to this principle while realizing little idle and throughput time. Prior to discussing an automated layout of the taktline system, the basic concept will be clarified in detail with regard to objective dimensions as well as boundary criteria, taking into account the groundwork of Ziskoven.

2.1. Objective Dimensions of the Taktline System

In the following, the order of necessary technological operations of a job will be called *process sequence*. Machines (including manual work stations), on which these technological operations are conducted, will be denoted by generic letters form the alphabet. Further, a taktline constists of severeal ordered taktstations. These taktstation may contain several taktsubstations, which are in fact, machines.

A taktline should be chosen such that similar process sequences can be manufactured on it. See Figure 2 for an exemplary illustration of three jobs, their corresponding process sequences (job 1: B-C-D-E; job2: D-F-G-I; job 3: A-C-D-H-I) as well as a taktline with six taktstations (A/B-C-D-F-G/H-I).

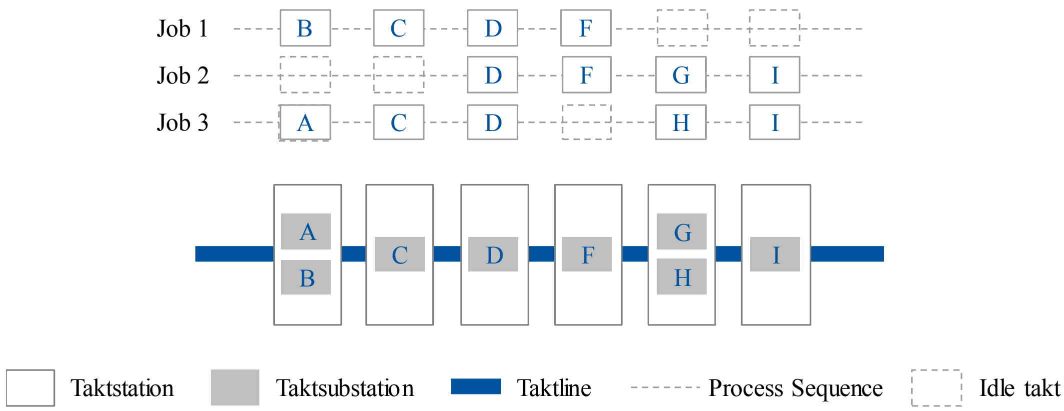

Fig. 2 - Illustration of taktstations, taktsubstations and idle takts

Evidently, the process sequences are not equal, but bear similar sections so that a consolidation thereof to one taktline seems reasonable. In this constellation each jobs' process sequence contains so-called idle takts. In the context of synchronized job shop production, e.g. job 3 is transported (as a bundle with job 1 and 2) along the taktline. At taktstation four ("F"), job 3 is not processed but would wait for a takt. This *prolongation* of jobs contradicts generally the goal of minimum through put times. However, it enables for finding a limited number of taktlines so that the advantages regarding operability and transparency of a synchronized manufacturing can be realized. In addition to the matter of idle takts, the above-illustrated taktline contains taktsubstation within the taktstations. Allowing for taktsubstations also enables finding a handable number of taktlines, while covering a large proportion of jobs' process sequences (Ziskoven, 2013). The concept of taktsubstations resembles the idea of different technologies within cell manufacturing (Murugan, 2011). Ideally, the machine 'G' and 'H' at taktstation five are either similar in their technology or required expertise for they could make up a taktstation. Additionally, they are ideally geographically close to each other on the shopfloor. If neither technological or geographical proximity is given, it could still be reasonable to consolidate them to one taktstation if they are complementary in their technology so that many job pass either machine. Within the concept of the taktline finding, different (mutual) distances of machines will be considered i.t.o. a superordinate key figure, which combines the above mentioned proximities.

See Figure 3 for an illustration of the dependencies of all three objective criteria. Obviously, the improvement of one objective generally implies the degradation of the other objective criteria. For explaining the connections, the example given above may be regarded.

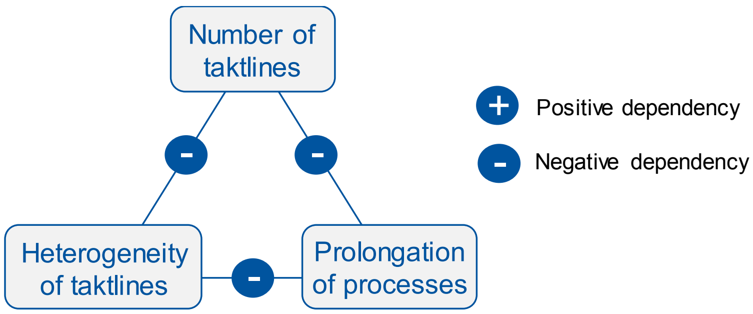

Fig. 3 - Dependencies of objective dimensions

For the mathematical representation of finding taktlines, a clustering analysis is applied. Clustering algorithms aim at finding homogeneous groups in large data sets given specific comparative criteria (Guojun, 2007). The comparative criteria should characterize the set's items most meaningful for a problem at hand. In the context of production, clustering algorithms have predominantly been applied for furnishing manufacturing cells (Murugan, 2011), e.g. for grouping technologies, which is related to the subject of finding taktlines for job shop production to a certain degree. Yet here, the order of a job's process steps (technologies a job passes) is crucial. Therefore, the process sequences as such need to be regarded as comparative criteria. The process sequences can be expressed as strings, i.e. ordered sets of elements. Clustering of strings is also referred to as sequence clustering (Li, 2006) and will be considered in this research.

In the following, a proper (string) distance measure for finding taktlines is developed, which takes into account the distance between items due to the process sequences' order as well as mutual distances between machines. A partitioning cluster algorithm is further developed to iteratively obtain a global minimal distance regarding the taktline system. Results of the proposed analysis will be displayed in section 4.

3. CLUSTER ANALYSIS

Cluster analysis is a meta-method for unsupervised classification of objects with regard to certain characteristics (Jain, 1999). The objective of a cluster analysis is to allocate objects into groups such that the objects within a group are similar to each other (homogeneous), and dissimilar to each other (heterogeneous) between groups (Guojun, 2007).

The core element of a cluster analysis is the definition of a distance (or similarity) measure for comparison of the objects (section 3.1). Further steps comprise of designing the cluster algorithm (section 3.2) and conducting the cluster validation (section 3.3). Refer to the next section for the specific cluster algorithm proposed for the finding taktlines.

3.1 Distance Measure

The distance measure is essential for the representation of the problem (Jain 1999). As described above, the process sequences can be represented as strings. Prior to discussion a proper string distance measure, the requirements will be illustrated. The process sequences of job 2 (DFGI) and 3 (ACDHI) for serve as an example. The distance between these two strings (process sequences) is a function of the objective dimensions 'taktline homogeneity' and 'prolongation'. The objective dimension of finding few taktlines is not considered within the distance calculation, since it represents the overall aim and is accounted for in the general problem statement.

In literature, numerous well-known string distant measure exist, e.g. the Hamming Distance (Hamming, 1950), Levenshtein Distance (Levenshtein, 1966) or Longest Common Subsequence (Hirschberg, 1977). Although a detailed description of these fundamentals distance measures will not be conducted, it is stated that the Longest Common Subsequence best matches the problem at hand, since the matching chacters do need to succeed each other (Hirschberg, 1977). Regarding the process sequences of job 2 and 3, (*A I)* would be the Longest Common Subsequence. The Longest Common Sequence does not, however, allow for a mismatch of which is necessary for the realization of the concept of taktsubstations. Therefore, besides the 'match' of characters (cost $c_M = 0$) or the insertion of an idle character (at the cost c_P), the possibility of a 'mismatch' (of cost $D12$) is given, which renders a taktsubstation in this context.

Let the cost for prolongation be $c_P = 2$ per idle takt and let the mutual distances for all machine pairs be as displayed in Table 1.

Table 1 - Mutual Distances between Machines

M1	A	A	A	A	A	A	C	C	C	C	C	D	D	D	D	F	F	F	G	G	H
M2	C	D	F	G	H	I	D	F	G	H	I	F	G	H	I	G	H	I	H	I	I
D12	2	3	2	2	2	1	3	1	2	2	2	1	2	1	4	2	3	2	1	2	4

The minimum distance between the process sequences of job 2 and 3 is then obtained as illustrated in Figure 4.

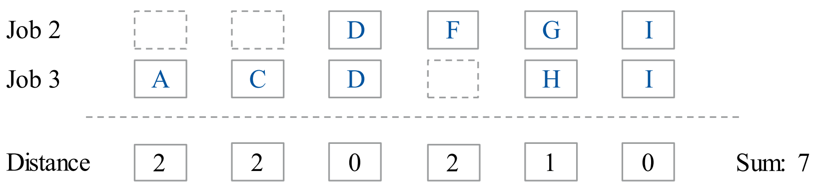

Fig. 4 - Distance Calculation for Two Strings

The implementation of this distance measure was done by means of a Needleman-Wunsch Algorithm, which is in turn based on Dynamic Programming (Sung 2010, Bergroth 2000).

The Needleman-Wunsch Algorithm allows for mismatches during the calculations of the optimal alignment and thus meets the requirements from above.

3.2 Cluster Algorithm

There are different approaches to grouping items to clusters. The most common differentiation is by dividing cluster algorithms into partitioning vs. hierarchical methods (Guojun, 2007). Hierarchical methods iteratively compose items to clusters (agglomerative hierarchical clustering) or disassembly (divisive hierarchical clustering) larger sets of items into clusters, by composing the currently most similar items or cluster together (or by disassembling to most heterogeneous cluster for the divisive method, respectively) (Xu, 2005). Partitioning clustering algorithms, on the other hand, groups items into a predefined number of clusters (Xu, 2005). During the course of the partitioning cluster algorithm, initial cluster elements are chosen, whereas the remaining items are allocated to the nearest cluster. A full permutation of the initial cluster elements ensure the finding of the optimal solution (Xu, 2005).

The advantage of partitioning clustering algorithms lies in the optimality, which is reached during the calculation. Hierarchical algorithms bear the risk of terminating in a local optimum (Xu, 2005). In the context of the taktline finding, however, optimality is important for the sake of efficiency and reproducibility of the solution if given slightly differing input data. Hence, a partitioning cluster algorithm was implemented for the taktline finding. However, for the estimation of the number of underlying cluster, a hierarchical clustering algorithm was conducted prior to the partitioning clustering. Even though not optimal, the results of the hierarchical clustering - usually presented in a so-called dendrogram - serve as a valid indicator for the number and structure of underlying homogeneous groups in a data set.

3.3 Cluster Validation

A cluster analysis always renders a result, the logical interpretations of the outcome - or rather, the verification of inherent clusters within the (input) objects needs to be considered within the context of the analysis. There is no general approach to proof the existence of clusters but it requires expertise (Jain, 1999). Nonetheless, the issue can usually be supported by e.g. visual measures (Weinan, 2002). Whereas a visualization will be provided when presenting the results, a theoretical discussion will be conducted in this section.

The existence of clusters within different jobs' processing sequences are highly probable. Although products in tool manufacturing are individual, some processing orders will be similar for a great amount of products. For example, sawing - drilling - eroding is a typical chain of processes conducted on a job at the beginning of the manufacturing, whereas hardening or polishing are required towards the end. These ordered processing steps are based on technological necessity and thus foster the presence of clusters within a set of processing steps.

In the next section, the proposed cluster analysis will be conducted and evaluated in terms of a proof of concept. Hereafter, a discussion and outlook to further research follows in the conclusion.

4. RESULTS

For the illustration of results, real-world data from a tool manufacturer were applied. The original data was codified for the sake of confidentiality. In total, 49 process sequences were considered for the cluster analysis, comprising of a total of 41 different machines. In Figure 5, the process sequences are displayed in descending frequency.

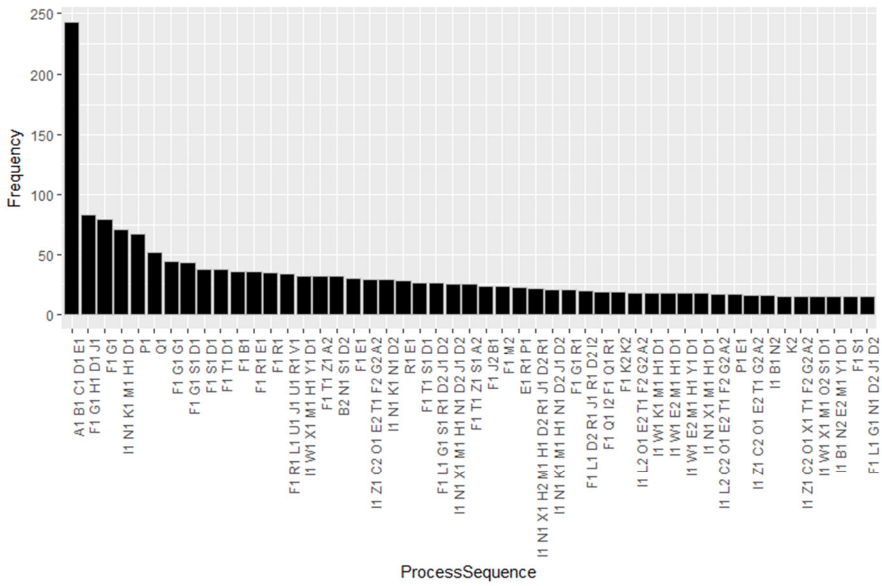

Fig. 5 - Barplot of Process Sequences in Descending Order

For the realization of the cluster analysis proposed above, mutual distances between machines need to be set in the context of taktsubstations. Further, the cost for prolongation (idle takts) needs to be settled. For the proof of concept within this research, the mutual distances were randomly set to other 2 or 4, whereas the cost for an idle takts was set to 3. By this choice, a unambiguous preference towards either a taktsubstation or prolongation is rendered during the calculation. Preparatory to the partitioning cluster algorithm, the distance matrix for all process sequences was calculated by means of the Needleman-Wunsch Algorithm as proposed in section 3.1. Afterward, for estimating the underlying number of cluster, a hierarchical cluster algorithm was pursued, rendering a dendrogram (see Figure 6). To this end, the open source software RStudio was applied (RStudio).

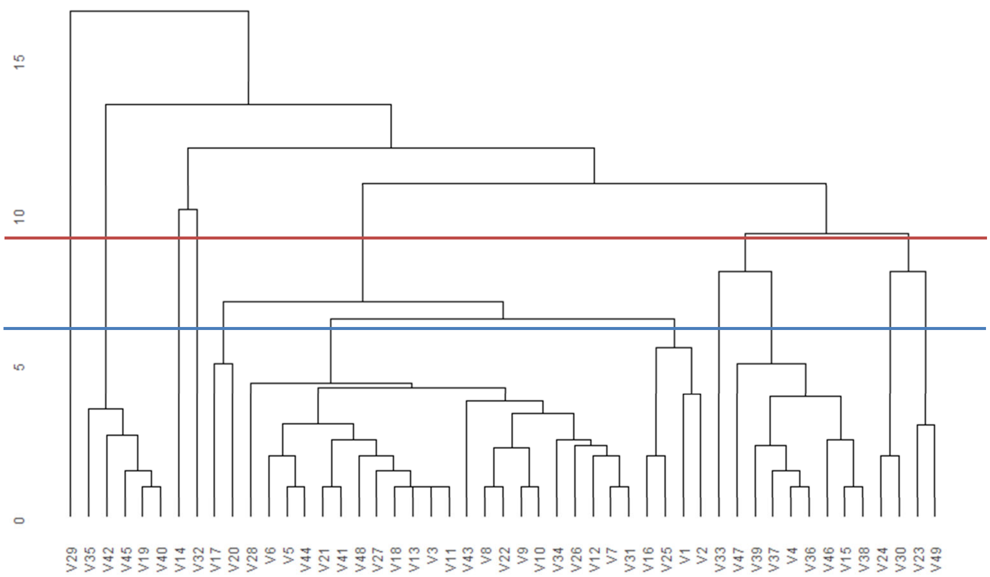

Fig. 6 - Dendrogram of Hierarchical Clustering of Process Sequences

The values on the vertical axis represent the distances; the horizontal axis on the other represents the input data. The numbers (V1-V49) corresponds to the order of process sequences in Figure 5. Depending on the sought number of taktlines the number of clusters was chosen i.t.o. a 'cut-off' distance regarding the dendrogram. Two scenarios were conducted during this research. In the first scenario, the cut-off distance was set to 9 (rendering ca. 7 clusters, see the red line); whereas in the second scenario, the cut-off distance was set to 5.5 (rendering ca. 11 clusters, see the blue line). A continuous variation for a 'number of taktlines'/'homogeneity of taktlines' portfolio could generally be worked out for a fully transparent solution space. This is, however, not part of this paper.

The partitioning cluster algorithm was implemented in C# (Visual Studio, 2015) and conducted for the two scenarios drawn above. The target value of the algorithm was the sum of squared distances over all clusters. For scenario one, seven clusters were obtained, with a total sum of squares of 658 (average 94), whereas for the second scenario, eleven clusters were obtained (371 sum of squares, 33.8 in average). Obviously, the heterogeneity within the taktlines is larger for fewer lines.

In accordance with the preparatory hierarchical clustering, for the two scenarios, similarly sized clusters were obtained. In order to gather the taktlines, a so-called traceback matrix was generated during the Needleman-Wunsch Algorithm. The traceback matrix reflects the actual best alignment, whereas the Needle-Wunsch algorithm - as Dynamic Programming in general - solely renders the optimal target value (Chebil, 2015).

See Figure 7 for an exemplary building of a taktline based on the underlying cluster. In the example displayed, two of the five process sequences bear idle takts, whereas two of the nine taktstations comprise of taktsubstations.

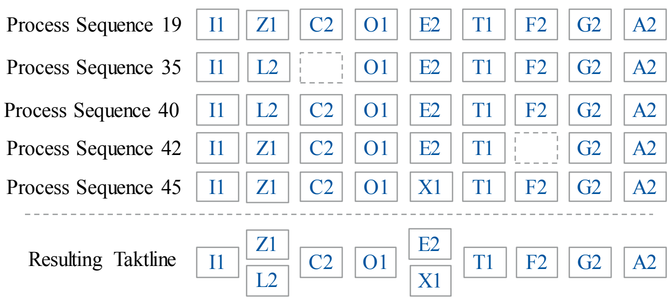

Fig. 7 - A Cluster's String Alignment for Generating a Taktlines

This alignment was executed for all clusters within the scenarios. See Table 2 and Table 3 for the overall results, summed up per scenario with regard to the objective dimensions.

Table 2 - Cluster Results for Scenario 1

Scenario 1	Length (number of taktstations)	Including process sequences	Average taktsubstations (per taktstation)	Average idle takts (per taktstation/process sequence)	Sum of squared distances
Taktline 1	8	9	2	0.21	90
Taktline 2	6	28	5.6	0.5	451
Taktline 3	8	1	1	0	0
Taktline 4	9	5	1.2	0.04	15
Taktline 5	11	1	1	0	0
Taktline 6	9	3	1.3	0.074	53
Taktline 7	10	2	1.1	0.2	49
Average (taktline)	**8.7**	**7**	**1.88**	**0.15**	**94**

Table 3 - Cluster Results for Scenario 2

Scenario 1	Length (number of taktstations	Including process sequences	Average taktsubstations (per taktstation)	Average idle takts (per taktstation/ process sequence)	Sum of squared distances
Taktline 1	8	8	1.75	0.188	54
Taktline 2	5	24	4.6	0.48	249
Taktline 3	8	1	1	0	0
Taktline 4	9	5	1.22	0.044	15
Taktline 5	5	1	1	0	0
Taktline 6	8	2	1.125	0.06	9
Taktline 7	5	3	2	0.067	40
Taktline 8	11	1	1	0	0
Taktline 9	9	2	1.11	0	4
Taktline 10	8	1	1	0	0
Taktline 11	6	1	1	0	0
Average (taktline)	**7.45**	**4.45**	**1.53**	**0.076**	**33.7**

As was assumed in section 3, the objective dimensions contradict each other. Thus, a smaller number of taktlines renders (in average) more heterogeneous taktlines, more idle takts and larger number of taktsubstations. Nevertheless, the general approach seems appropriate for finding a set of taktlines. The distance measure is adaptive and fulfills the requirements regarding idle takts and taktsubstations. The partitioning cluster algorithm renders different, but comparable results regarding the clusters' content as such, when varying numbers of clusters are induced.

5. CONCLUSION

In this paper, a clustering algorithm for finding taktlines in the context of synchronized job shop production was proposed. To this end, a string distance measure was presented embedded in a partitioning clustering algorithm. On the basis of real-world data, the proof of concept was conducted. The assumption and outline seem reasonable on the basis of obtained results.

As a drawback, the number of obtained taktlines (or the homogeneity of taktlines, respectively) was not yet satisfying, given a rather limited number of (input) process sequences. Future research thus needs to address the possibility of logistic operations within the concept of the taktline finding. That is, taktlines do not necessarily have to represent whole process sequences but can be frequently reoccurring *chunks* of process sequences. By these means, fewer and more homogeneous taktlines could possibly be found.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the funding by the German Federal Ministry of Education and Research (BMBF) within the Program "Innovations for Tomorrow's Production, Services, and Work" (funding number 02P14B145)

REFERENCES

[1] Bergroth L, Hakonen H, Raita T. A Survey of Longest Commong Subsequence Algorithms: September 27-29, 2000, A Curuña, Spain proceedings. Seventh International Symposium on String Processing and Information Retrieval, pp. 39-48, 2000.

[2] Chebil K, Khemakhem M. A dynamic programming algorithm for the Knapsack Problem with Setup. Computers & operations research, 64, pp. 40-50, 2015.

[3] Eversheim W, Schuh G. Produktion und Management: Gestaltung von Produktionssystemen. Band 3, Hütte Produktion und Management. Berlin: Springer, Lehrstuhl für Produktionssystematik am WZL, 1999.

[4] Günther HO, Tempelmeier H. Produktion und Logistik. 6. Auflage. Berlin Heidelberg New York: Springer, 2012.

[5] Guojun G, Chaoqun M, Jianhong W. Data Clustering: Theory, Algorithms, and Applications. Philadelphia, PA, USA: ASA SIAM, American Statistical Association and the Society for Indsutrial and Applied Mathematics, 2007.

[6] Gruß R. **S**chlanke Unikatfertigung: Zweistufiges Taktphasenmodell zur Steigerung der Prozesseffizienz in der Unikatfertigung auf Basis der Lean Production. Springer Gabler, 2010.

[7] Hamming RW. Error Detechting and Error Correcting Codes. The Bell System Technical Journal, 1950 Nr. Vol. XXIX, No. 2, pp. 147-160, 1950.

[8] Humprey S. Mapping and Scheduling Algorithms for Synchronized Individual Production. Van Laack GmbH Aachen, 2016.

[9] Hirschberg DS. Algorithms for the Longest Common Subsequence Problem. Journal of the Association for Computing Machinery, Nr. Vol. 24, No. 4, pp. 664-675, 1977.

[10] Jain AK, Murty MN, Flynn PJ. Data Clustering: A Review. ACM Computing Surveys, 31 Nr. 3, pp. 264-323, 1999.

[11] Klotzbach C. Gestaltungsmodell für den industriellen Werkzeugbau. Shaker Verlag, 2007.

[12] Levenshtein VI. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady - Cybernetics and Control Theory, Vol. 10, No. 8, pp. 707-710, 1966.

[13] Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics (Oxford, England), 22 Nr. 13, pp. 1658-1659, 2006.

[14] Murugan M, Selladurai V. Formation of Machine Cells / Part Families in Cellular Manufacturing Systems Using an ART-Modified Single Linkage Clustering Approach, 2011.

[15] RStudio Team. RStudio: Integrated Development Environment for R. Boston, MA, 2015.

[16] Schmitt RH, Ellerich M, Humphrey S. Multi-objective allocation of customized orders to production-line networks. CIRP Annals, Volume 65, Issue 1, pp. 429-432, 2016.

[17] Sung WK. Algorithms in Bioinformatics: A Practical Introduction. London, England: Taylor & Francis Group, Chapman & Hall/CRC Mathematical and Computational Biology Series, ISBN 978-1-4200-7033-0. 2010.

[18] Weinan W, Osmar RZ. Clustering Web Sessions by Sequence Alignment. Proceedings of the 13th International Workshop on Database and Expert Systems Applications, 2002.

[19] Xu R, Wunsch D. Survey of clustering algorithms. IEEE transactions on neural networks, 16 Nr. 3, pp. 645-678, ISSN 1045-9227, 2005.

[20] Ziskoven H. Methodik zur Gestaltung und Auftragseinplanung einer getakteten Fertigung im Werkzeugbau. Apprimus Verlag Aachen, 2013.

[21] Zwanzig F. Taktung der Unikatfertigung am Beispiel des Werkzeugbaus. Aachen: Apprimus Verlag, Ergebnisse aus der Produktionstechnik, 2010.